



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πρόγραμμα Μεταπτυχιακών Σπουδών «Πληροφορική και Εφαρμογές»



Αρχές Ψηφιακής Τεχνολογίας

Γιάννης Βογιατζής
2018-2019

Επικοινωνία

Γιάννης Βογιατζής

voyageri@teiath.gr,

voyageri@uniwa.gr

Γραφείο Κ.16.207 (2^{ος} όροφος)

<http://eclass.uniwa.gr/>

Τελική Αξιολόγηση - Βαθμολόγηση

Τελικός Βαθμός

Βαθμός στην τελική εξέταση (70%)

Βαθμός εργασιών (30%)

Επισκόπηση

- Εισαγωγή στην ψηφιακή τεχνολογία, ιστορία των υπολογιστών
- Αναπαράσταση πληροφορίας στον υπολογιστή, συστήματα αρίθμησης, Αναπαράσταση αρνητικών αριθμών
- Πράξεις
- Παράσταση σταθερής και κινητής υποδιαστολής
- Οργάνωση υπολογιστή, ΚΜΕ, μνήμη, μέσα αποθήκευσης, μονάδες εισόδου εξόδου, δίαυλοι
- Άλγεβρα Boole και λογικές πύλες
- Ανάλυση και σχεδίαση συνδυαστικών κυκλωμάτων
- Στοιχεία μνήμης, ακολουθιακά κυκλώματα
- Ανάλυση και σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων
- Γλώσσες περιγραφής υλικού, VHDL vs Verilog
- Σχεδίαση συνδυαστικών κυκλωμάτων σε γλώσσα VHDL
- Σχεδίαση ακολουθιακών κυκλωμάτων σε γλώσσα VHDL

Τρανζιστορ

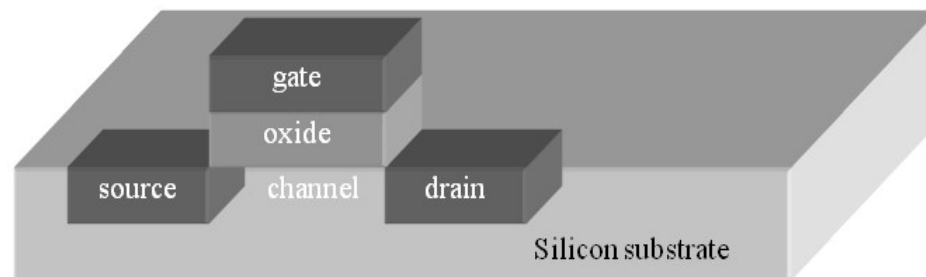
- Κατασκευάστηκε το 1948 στα Bell Labs
- In acknowledgement of this accomplishment, Shockley, Bardeen, and Brattain were jointly awarded the 1956 [Nobel Prize in Physics](#) "for their researches on semiconductors and their discovery of the transistor effect"



John Bardeen, William Shockley and Walter Brattain at Bell Labs, 1948.

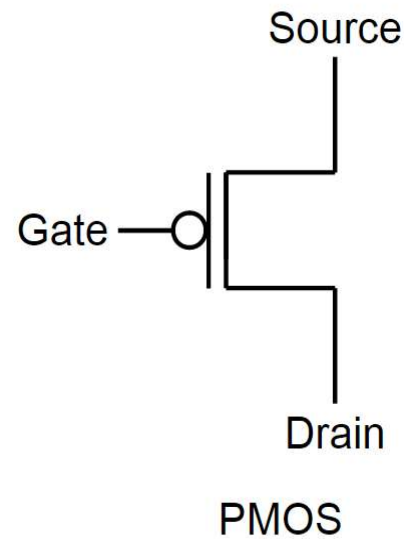
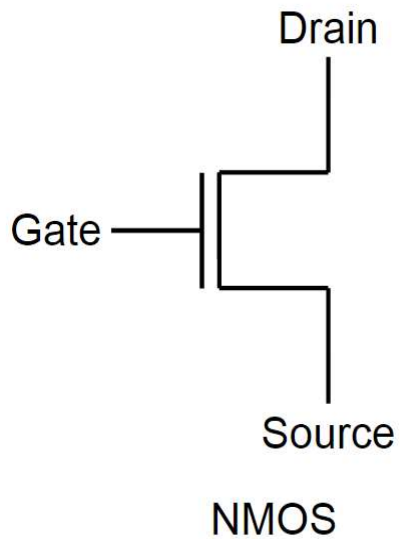
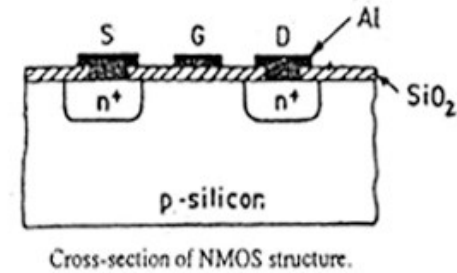
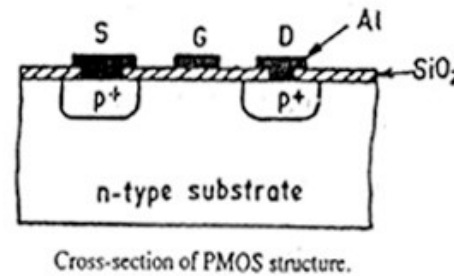
Λειτουργία Τρανζίστορ

- Σαν διακόπτης
- Επιτρέπει (ή όχι) τη διέλευση ρεύματος από το source στο drain ανάλογα με το ρεύμα στο gate



Λειτουργία Τρανζιστορ

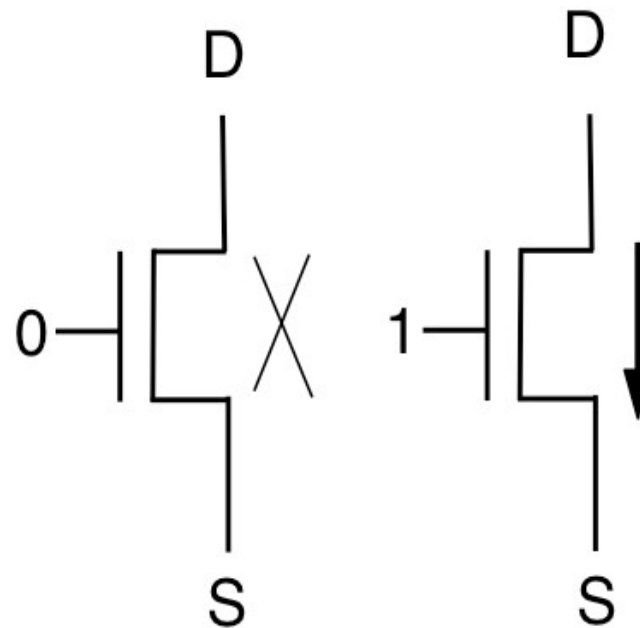
- Δύο τύποι: nMOS και pMOS



Λειτουργία Τρανζίστορ

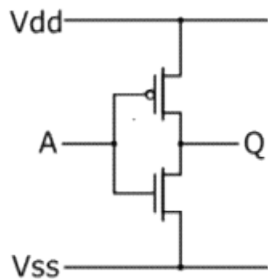
nMOS και pMOS

- nMOS: με 1 άγει, με 0 δεν άγει
- pMOS: με 0 άγει, με 1 δεν άγει



Τρανζίστορ και πύλες

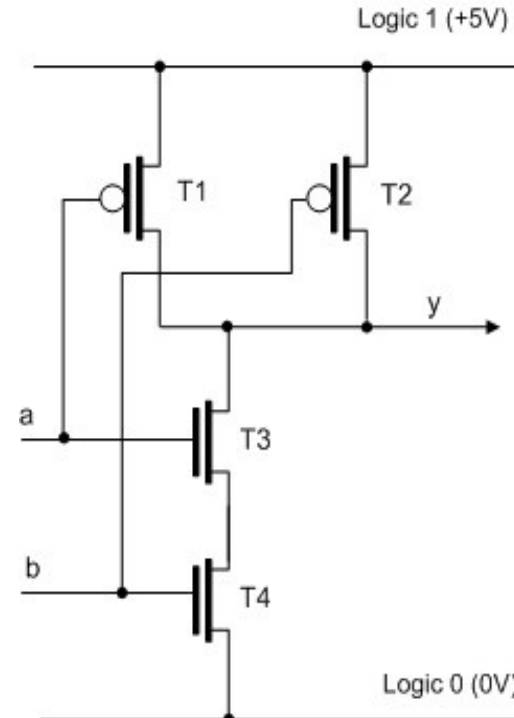
Συνδέοντας transistor κατάλληλα μπορούμε να φτιάξουμε πύλες ...



Πύλη NOT με transistor

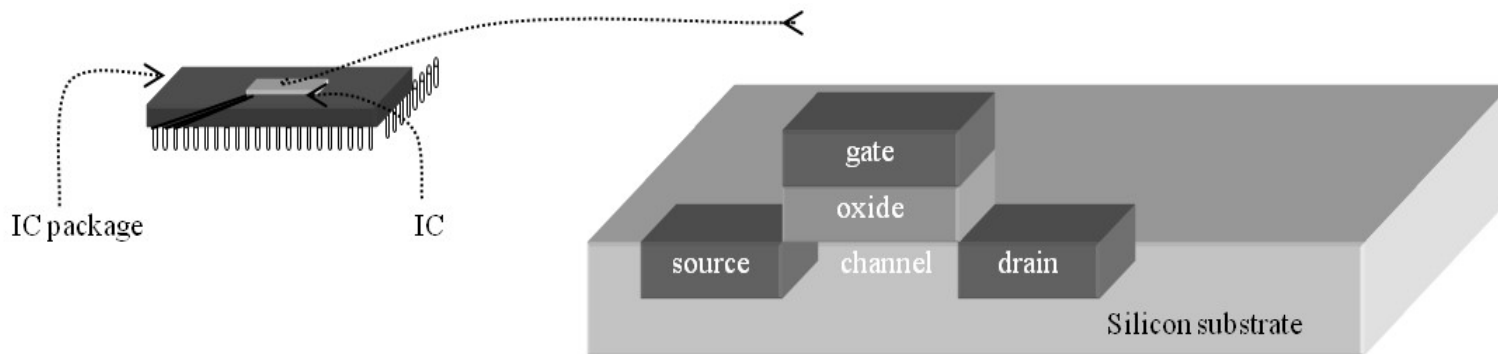
nMOS=> 1: A, 0: ΔA

pMOS=> 0: A, 1: ΔA



Τρανζίστορ και ολοκληρωμένο κύκλωμα

...και τελικά ολοκληρωμένα κυκλώματα



Τεχνολογία Κατασκευής (65 nm, 13 nm etc)

Νόμος του Moore

- Ο Gordon Moore προέβλεψε (1965, 1975) την συρρίκνωση του μεγέθους των transistors
- Διπλασιασμός της πυκνότητας ολοκλήρωσης (# transistors)
 - Κάθε 12 μήνες (1965)
 - Κάθε 24 μήνες (1975)

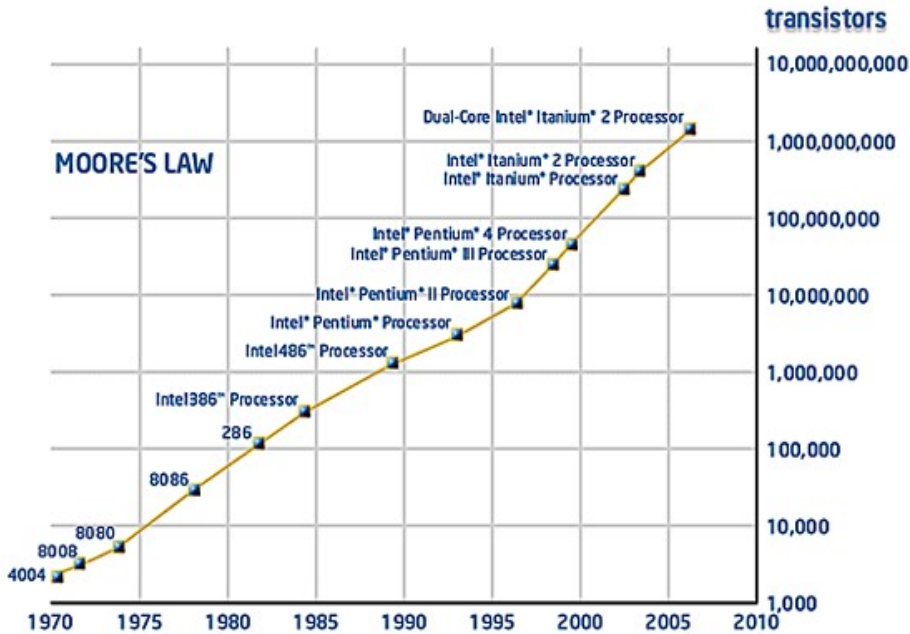


Gordon Moore Συνιδρυτής της Intel

Μια αναλογία

Αν οι αεροπορικές μεταφορές είχαν την ίδια εξέλιξη με αυτή του νόμου του Moore...

Το ίδιο ταξίδι θα κόστιζε σήμερα ένα **λεπτό του ευρώ** και θα διαρκούσε **1 δευτερόλεπτο!**



... και θα σταματούσε να λειτουργεί
μια φορά το χρόνο...



Το 1978 μια πτήση Νέα Υόρκη-Παρίσι κόστιζε **900 ευρώ** και διαρκούσε **7 ώρες**

Αποτελέσματα της εξέλιξης

- Αύξηση της απόδοσης
- Μείωση του κόστους
- Μικρότερα ολοκληρωμένα κυκλώματα (Ο.Κ.)
- Περισσότερες λειτουργίες

Φυσικά, υπήρξαν και λανθασμένες προβλέψεις...

**“I think there is a world market for
maybe five computers.”**

Thomas Watson,
Chairman of IBM, 1943

**“There is no reason for any
individual to have a computer in
their home.”**

Ken Olson,
President, Chairman and Founder of
Digital Equipment Corp., 1977

**“640K ought to be enough for
anybody.”**

Bill Gates, Microsoft founder, 1981
(though today he denies he said it)

Κατηγορίες Υπολογιστών



Επιτραπέζιοι
(Desktop)

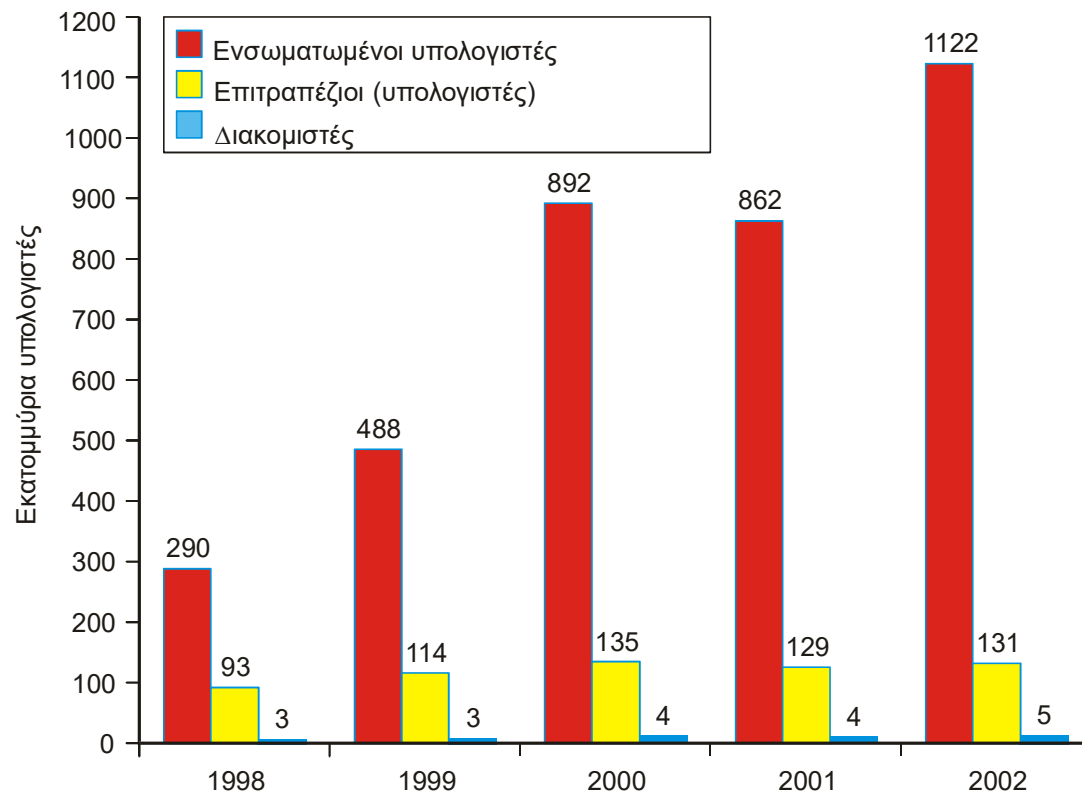


Διακομιστές
(Servers)

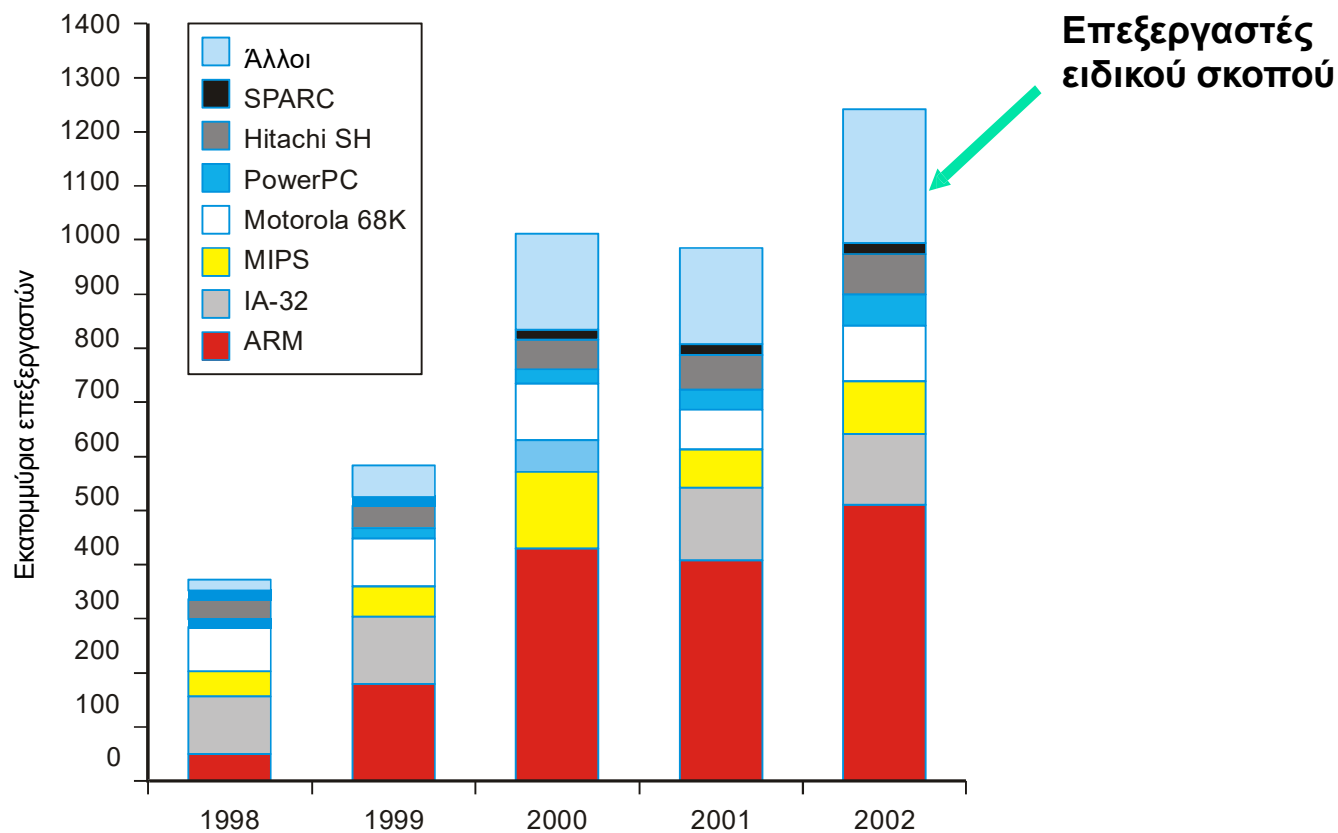


Ενσωματωμένοι
(Embedded)

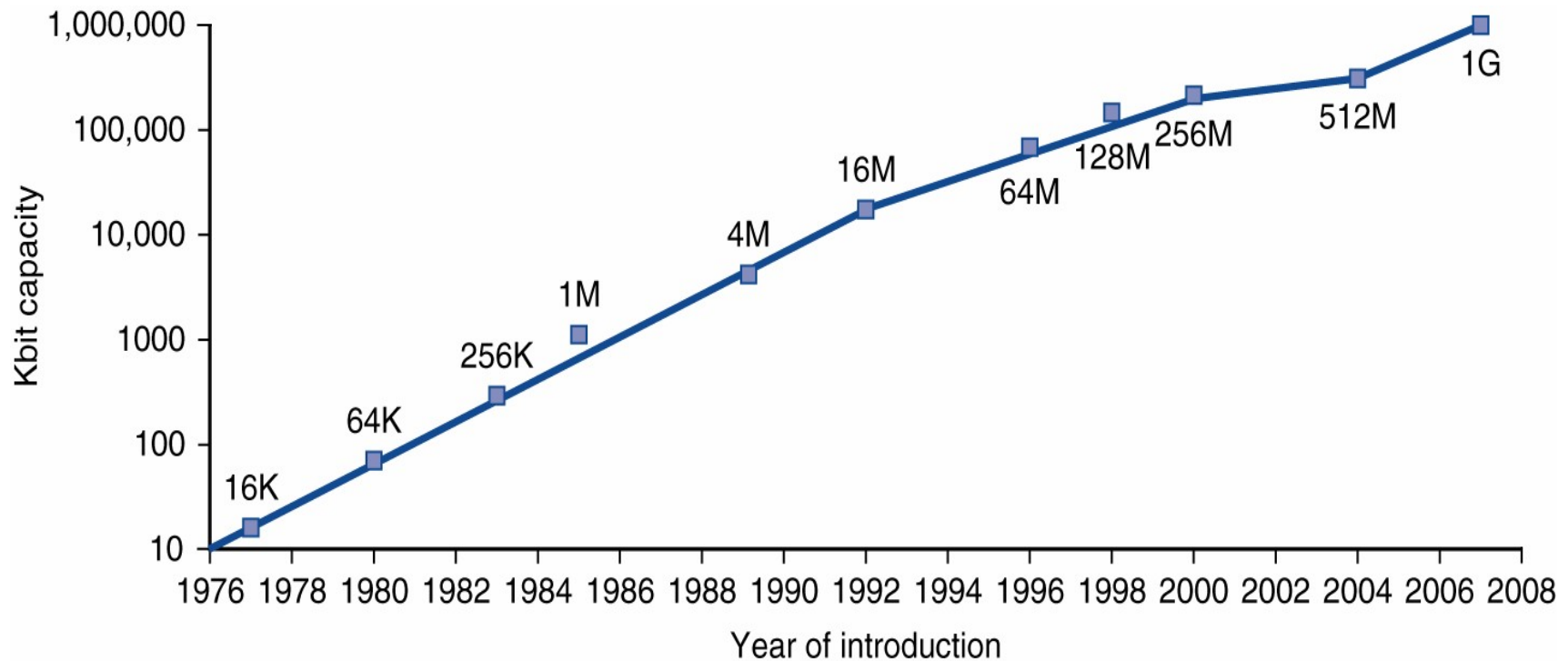
Αγορά υπολογιστών



Αγορά επεξεργαστών

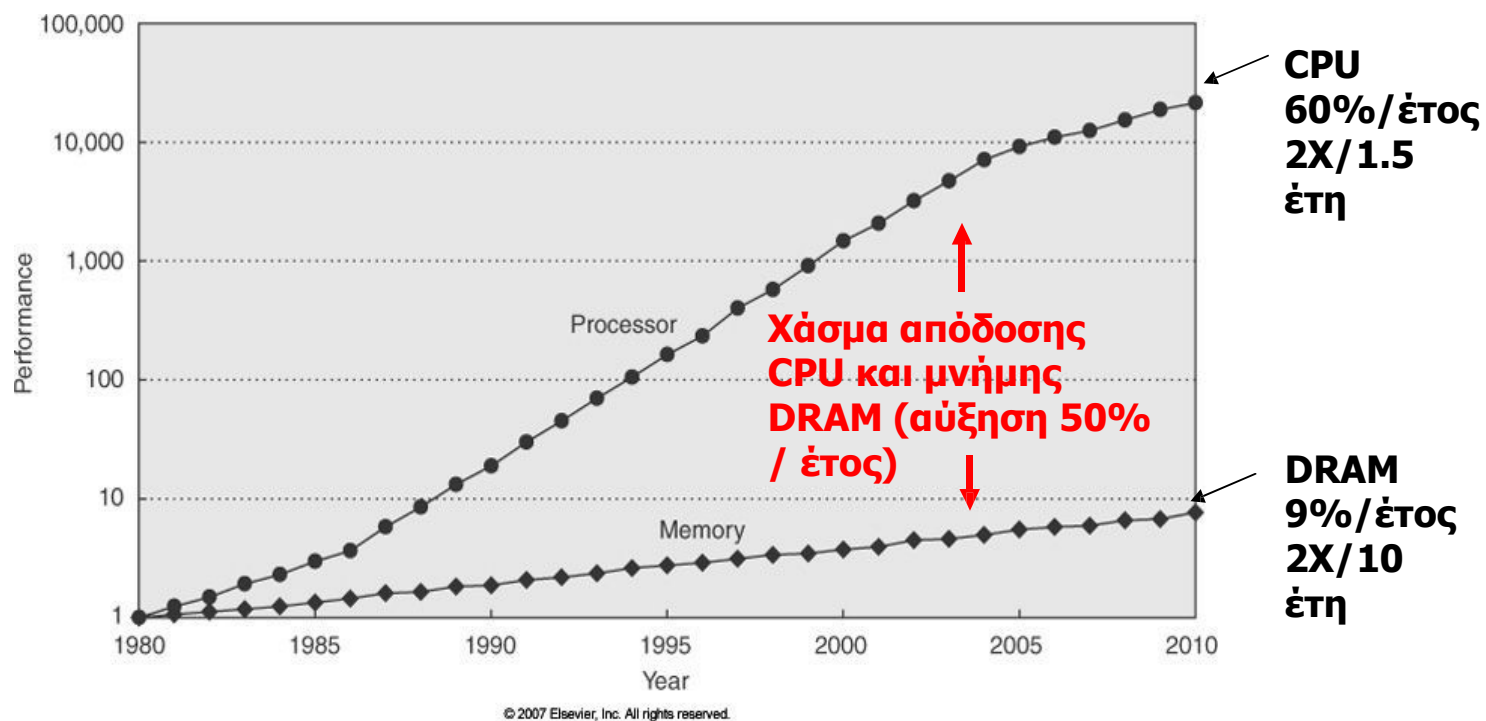


Χωρητικότητα DRAM

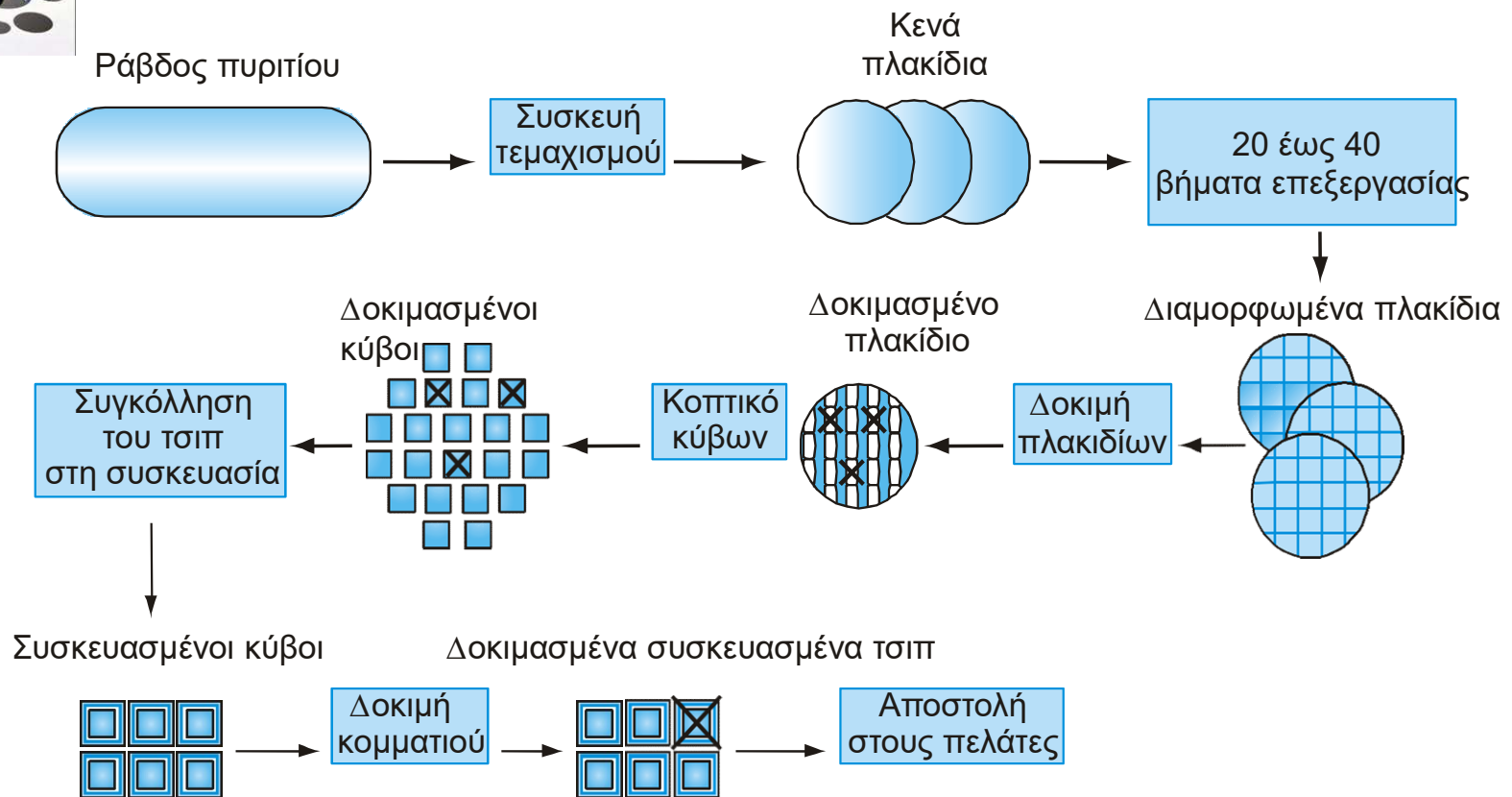


DRAM growth rule: 4x κάθε 3 χρόνια

Χάσμα απόδοσης CPU-μνήμης



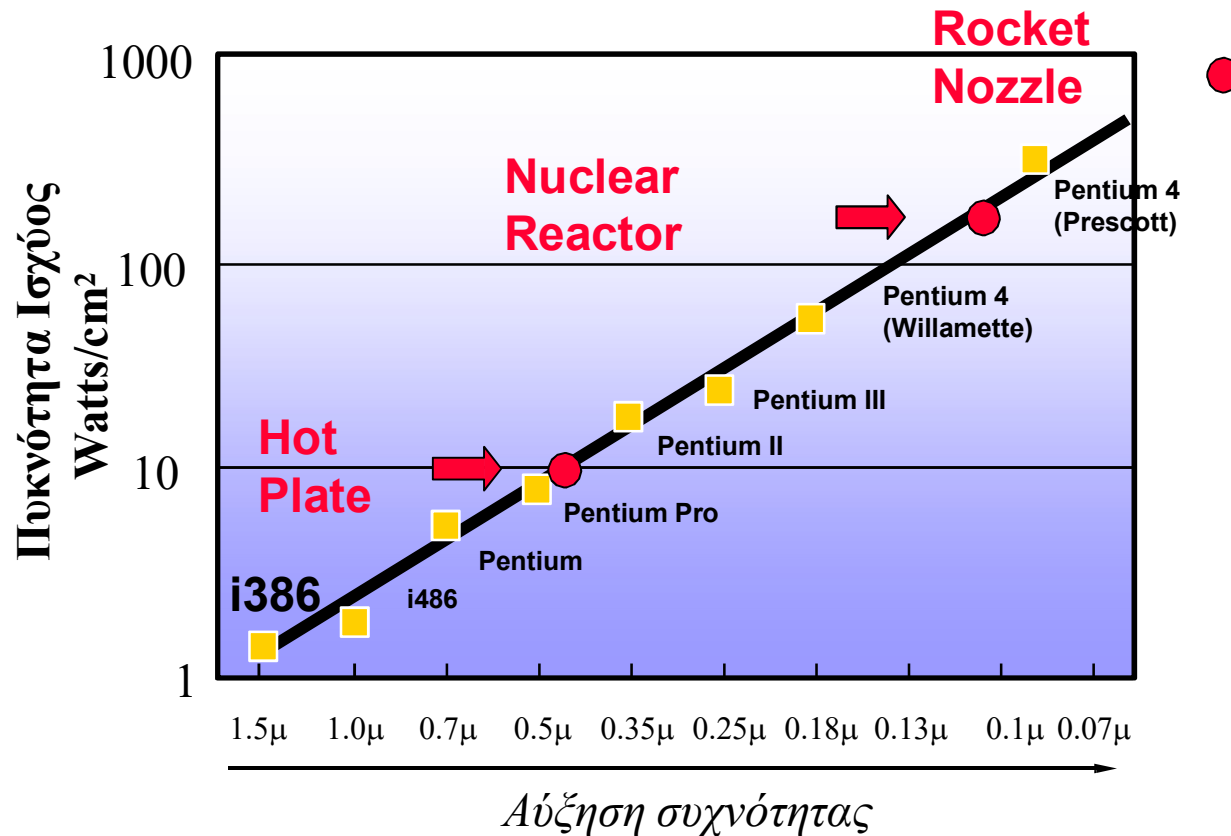
Διαδικασία κατασκευής ΟΚ (figure)



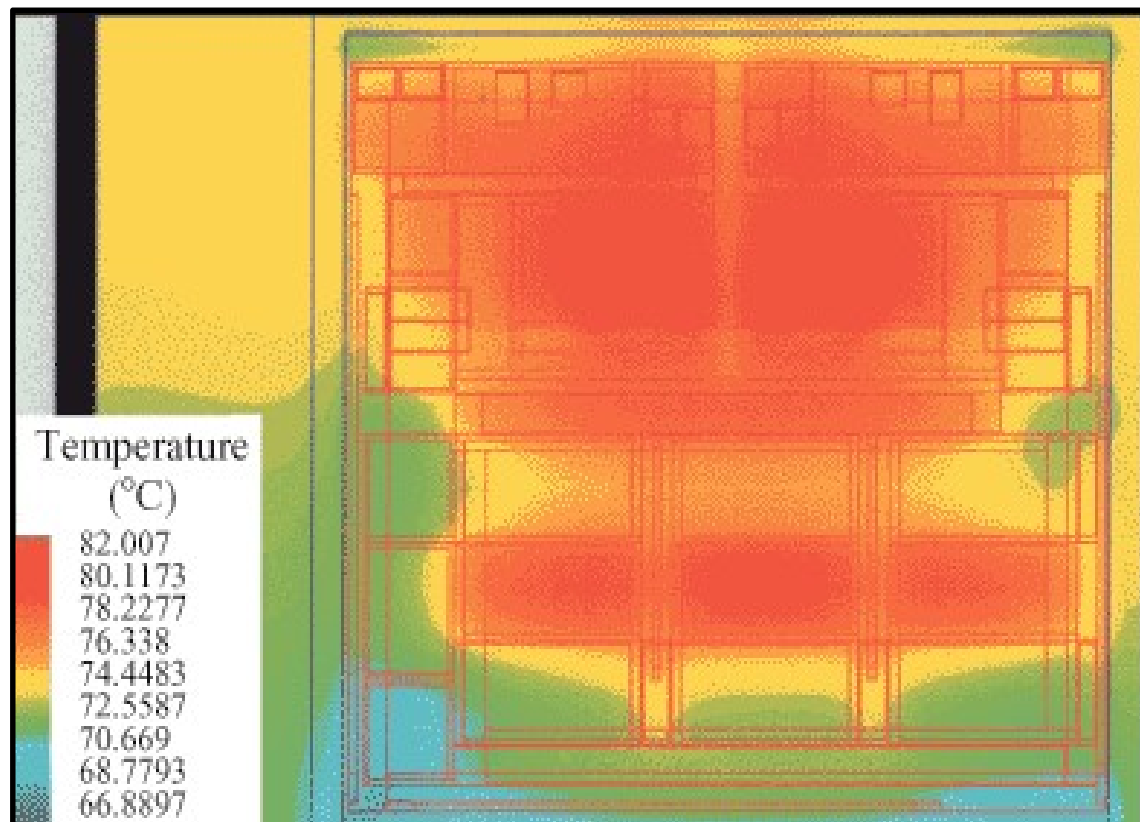
Μείωση Κατανάλωσης ενέργειας

- Μείωση του κόστους
 - Χρήση μικρότερης/φθηνότερης μπαταρίας
 - Χρήση φθηνότερης συσκευασίας O.K. (IC packaging)
- Αύξηση της αξιοπιστίας
 - Η αύξηση της θερμότητας μειώνει την αξιοπιστία
- Πρακτικοί λόγοι
 - Φορητές συσκευές μικρού μεγέθους και βάρους

Το πρόβλημα της θερμότητας στους σύγχρονους επεξεργαστές

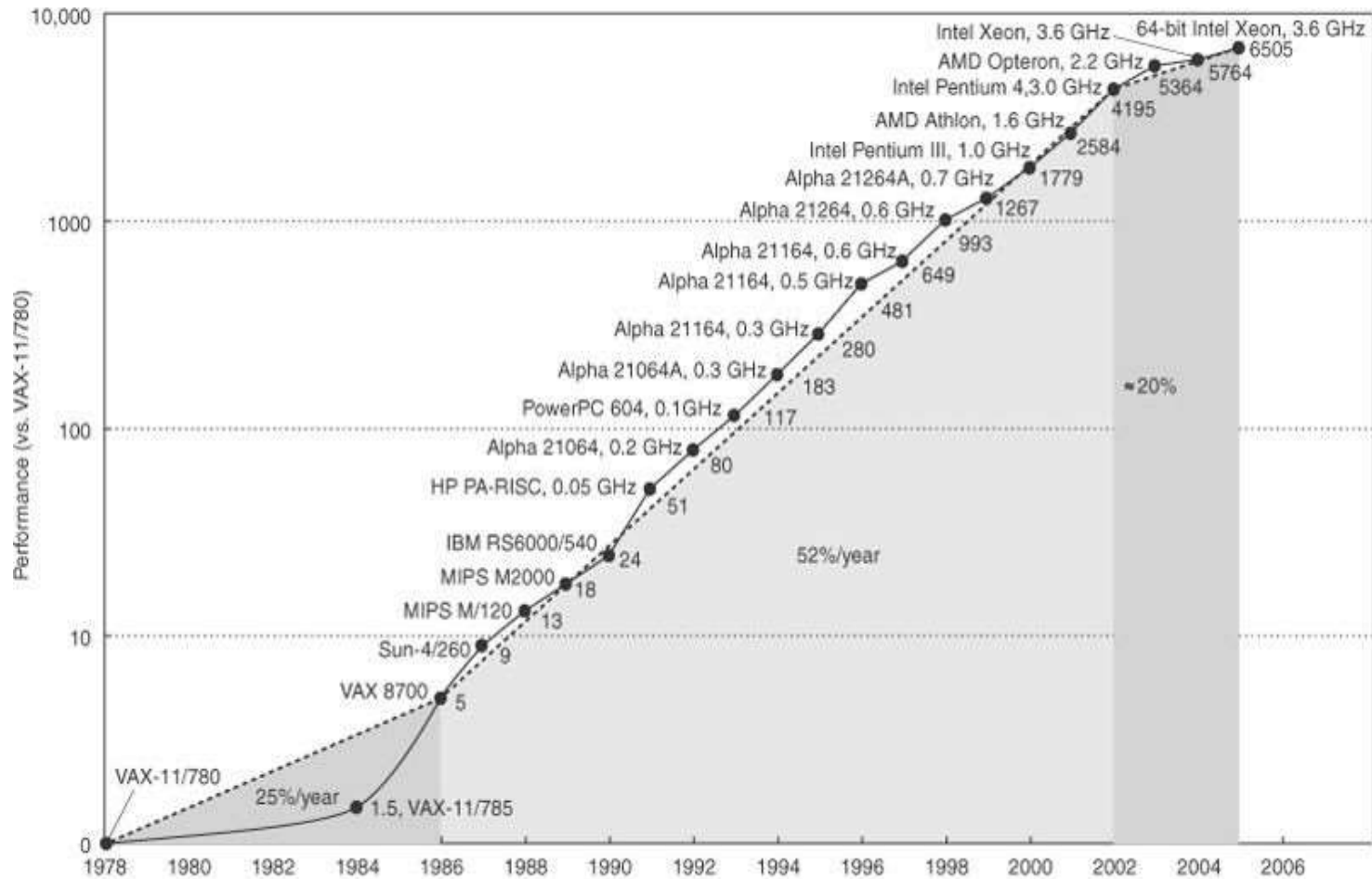


Πώς θερμαίνεται ο επεξεργαστής

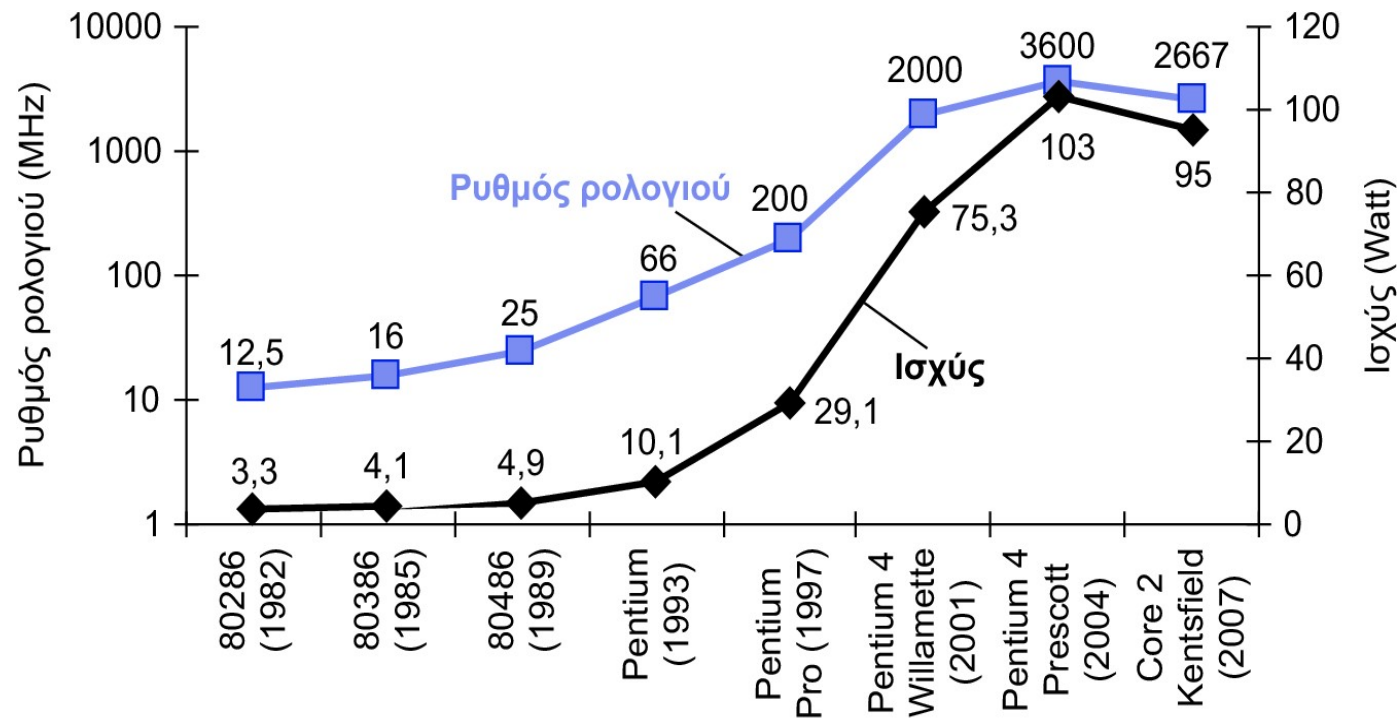


Θερμοκρασία: από 66.8 C μέχρι 82 C

Απόδοση επεξεργαστών



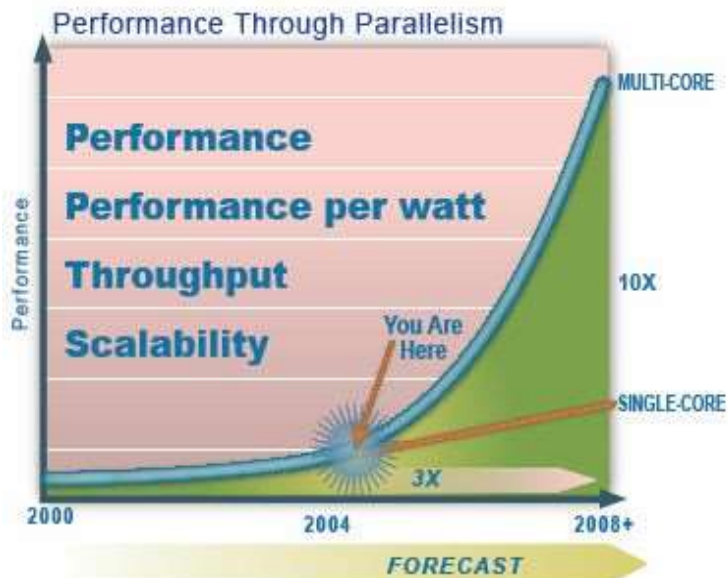
Η «κούρσα» για ψηλότερες συχνότητες



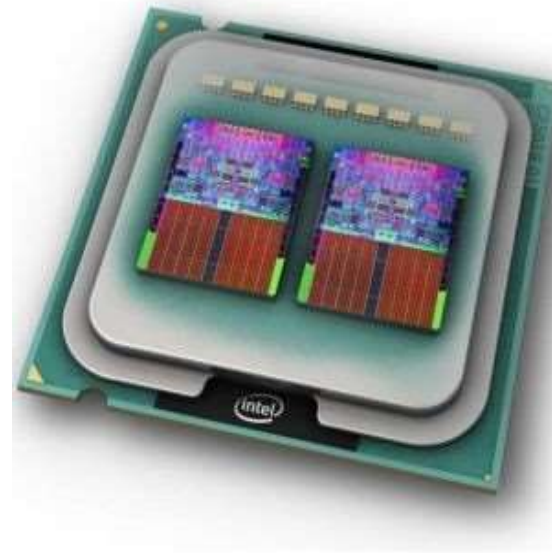
- Τι τη διέκοψε;
- Η αυξανόμενη κατανάλωση ισχύος και θερμότητας

Πολλαπλοί πυρήνες

- Πιο χαμηλή συχνότητα λειτουργίας
 - Πιο χαμηλή κατανάλωση ισχύος
- Αύξηση της απόδοσης
 - Απαιτείται κατάλληλο λογισμικό που να εκμεταλλεύεται την παραλληλία



Intel Quad-Core

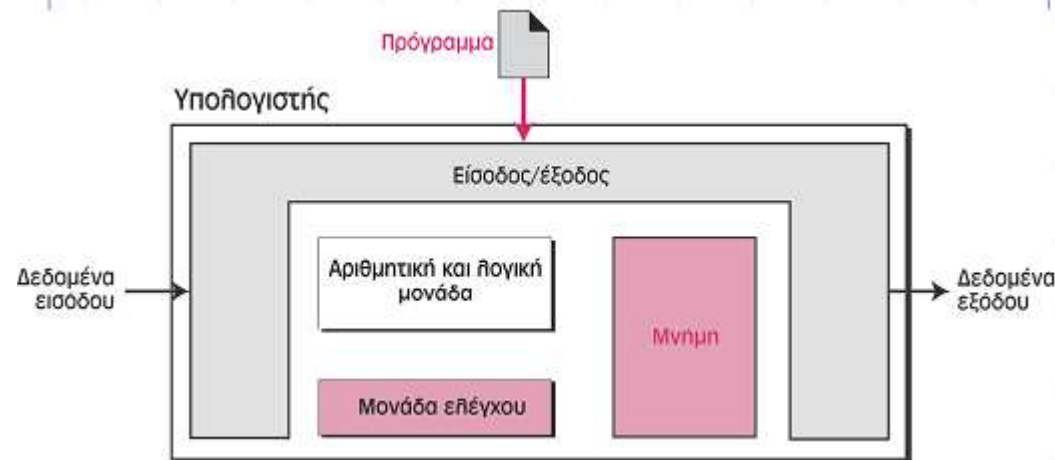


Πηγή: Intel

Μοντέλο Von Neumann (1950)

Ορίζει

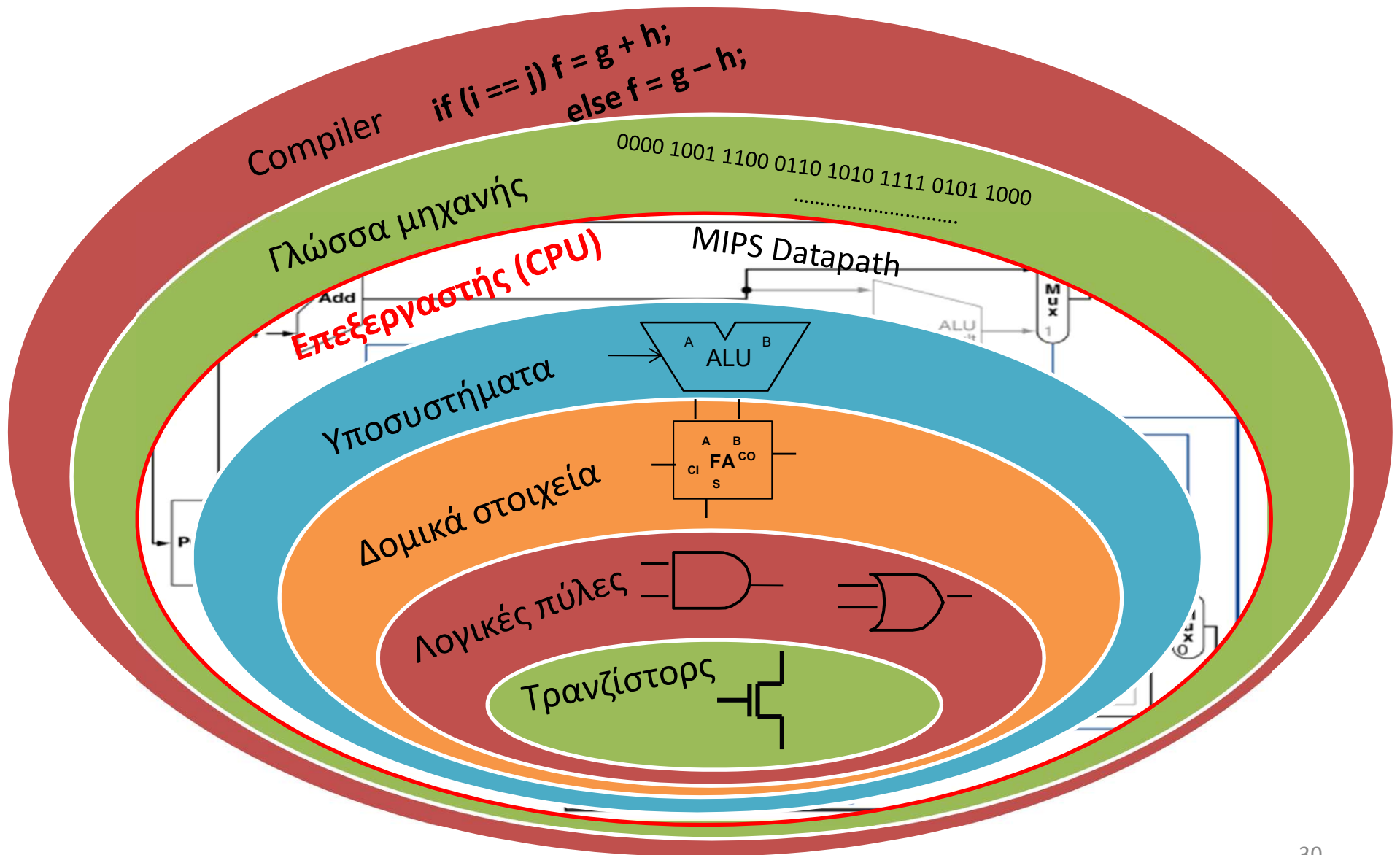
- τον υπολογιστή ως τέσσερα υποσυστήματα:
 1. Μνήμη
 2. Αριθμητική και Λογική Μονάδα
 3. Μονάδα Ελέγχου
 4. Είσοδος / Έξοδος
- ότι το πρόγραμμα πρέπει να αποθηκεύεται στη μνήμη.
- ότι το πρόγραμμα αποτελείται από πεπερασμένο αριθμό εντολών οι οποίες εκτελούνται **μία μετά την άλλη, σειριακά.**



Τι είναι υπολογιστής

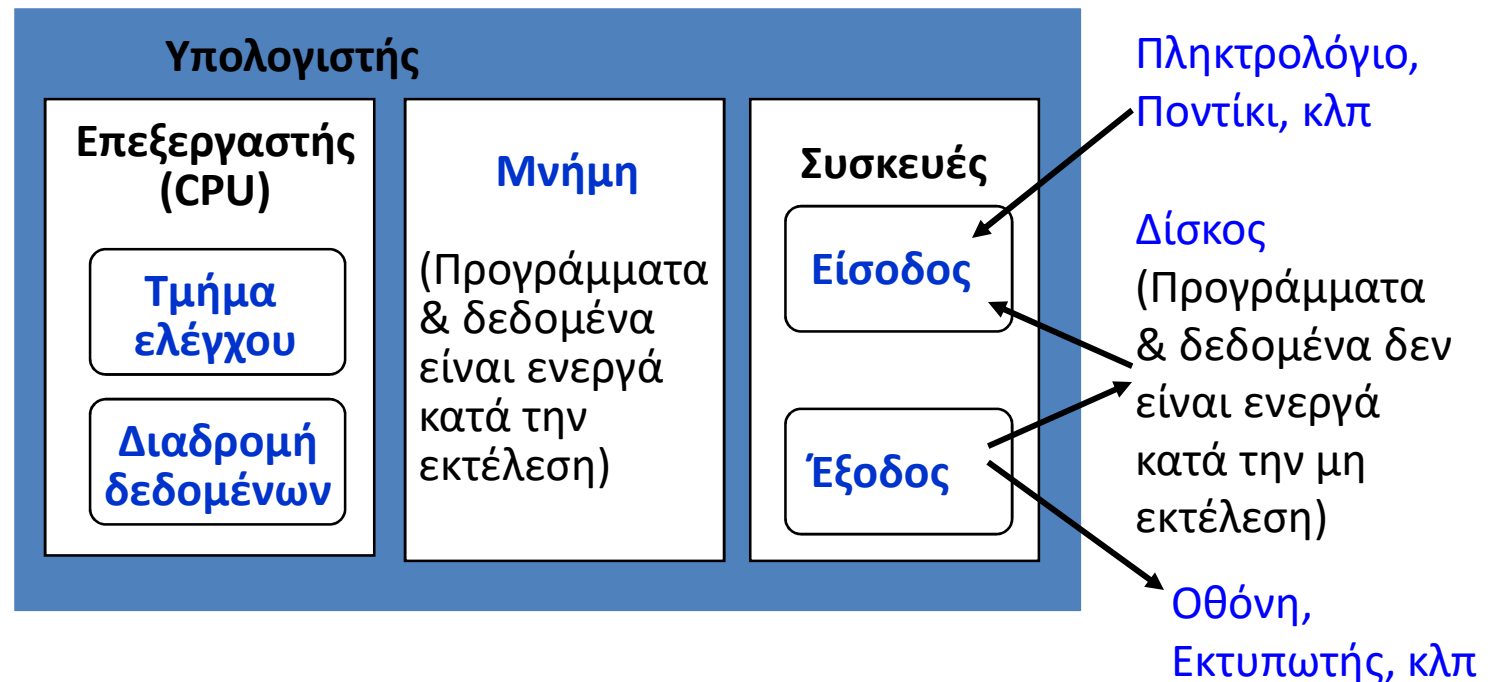
- Ένας υπολογιστής είναι ένα ψηφιακό σύστημα
 - Αποτελείται από συνδυαστική και ακολουθιακή λογική
- Τι κάνει αυτό το ψηφιακό σύστημα;
 - Κάνει ό,τι του λέμε να κάνει. Τίποτα περισσότερο, τίποτα λιγότερο
- Ένας υπολογιστής κάνει ακριβώς ό,τι του λέει να κάνει το λογισμικό
 - Λογισμικό είναι μια σειρά από εντολές
- Ένας υπολογιστής εκτελεί εντολές:
 - Αριθμητικές (πρόσθεση, αφαίρεση, διαίρεση,...),
 - Προσπέλασης στη μνήμη (ανάγνωση, εγγραφή),
 - Υπό συνθήκη (αν ισχύει μια συνθήκη ο έλεγχος του προγράμματος μεταβαίνει σε άλλο σημείο του προγράμματος), κλπ

Ένα υπολογιστικό σύστημα



Συστατικά τυπικού υπολογιστή

1. Τμήμα ελέγχου
2. Διαδρομή δεδομένων (καταχωρητές, ALU, δίαυλοι)
3. Μνήμη
4. Είσοδος
5. Έξοδος

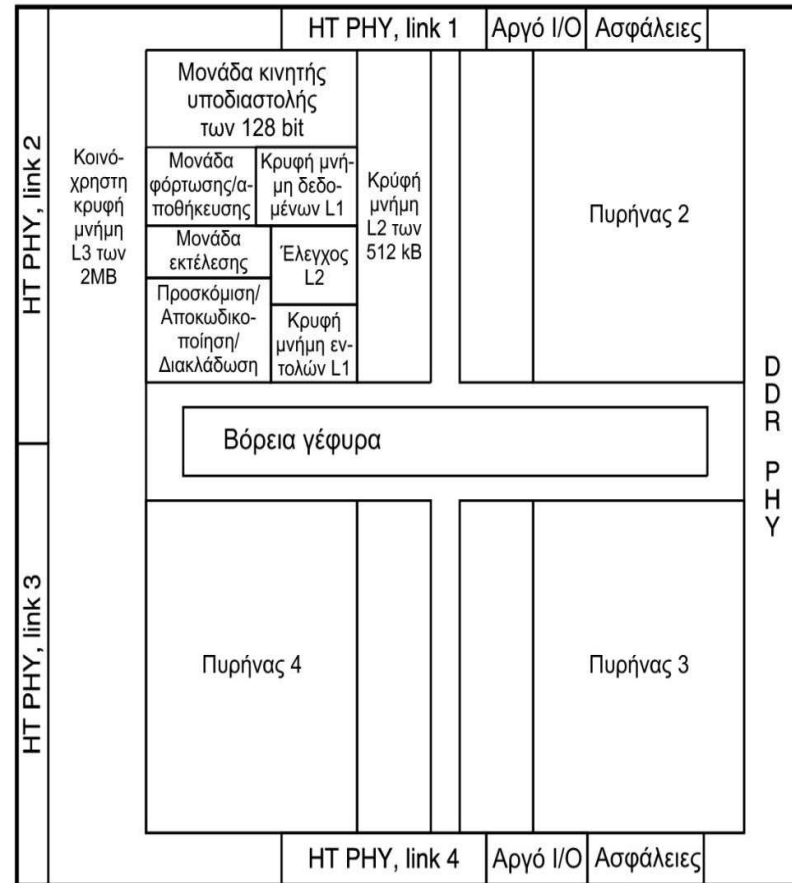
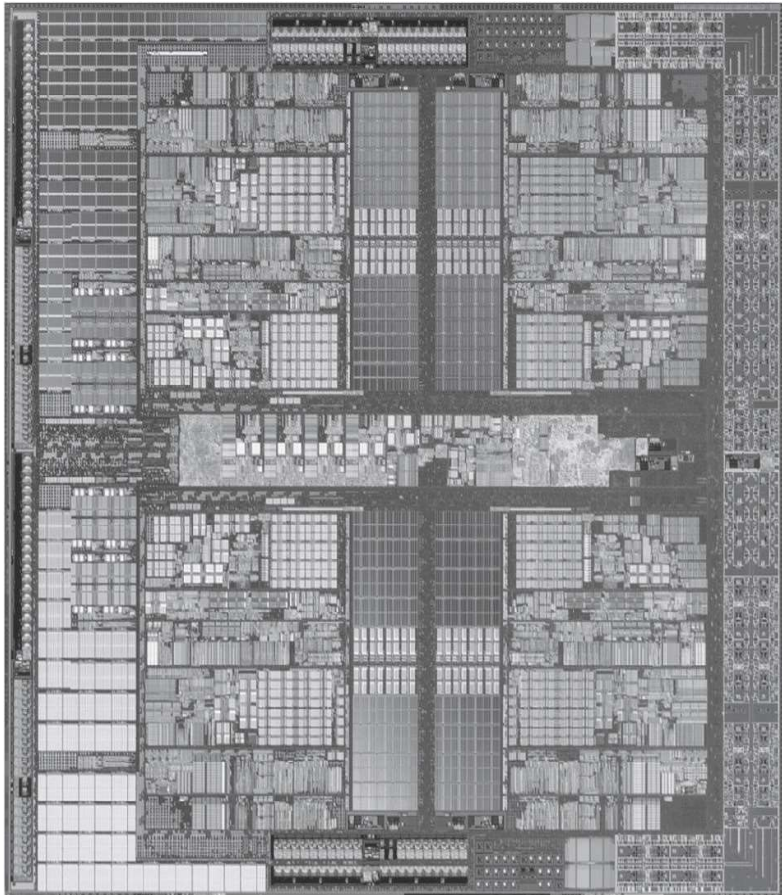


Συστατικά επεξεργαστή (CPU) (1/3)

- Μνήμες cache εντολών και δεδομένων
 - Μικρές και γρήγορες μνήμες που περιέχουν πρόσφατες εντολές/δεδομένα
- Αριθμητική και λογική μονάδα (Arithmetic & Logic Unit - ALU)
 - Εκτελεί αριθμητικές και λογικές εντολές
- Αρχείο καταχωρητών
 - Γενικού σκοπού καταχωρητές για υπολογισμούς
- Απαριθμητής προγράμματος (Program Counter, PC)
 - Περιέχει τη διεύθυνση επόμενης εντολής που θα προσκομιστεί
- Έλεγχος
 - Σήματα που καθορίζουν την εκτέλεση των εντολών
- Ρολόι: χρησιμοποιείται για χρονισμό

Συστατικά επεξεργαστή (CPU) (2/3)

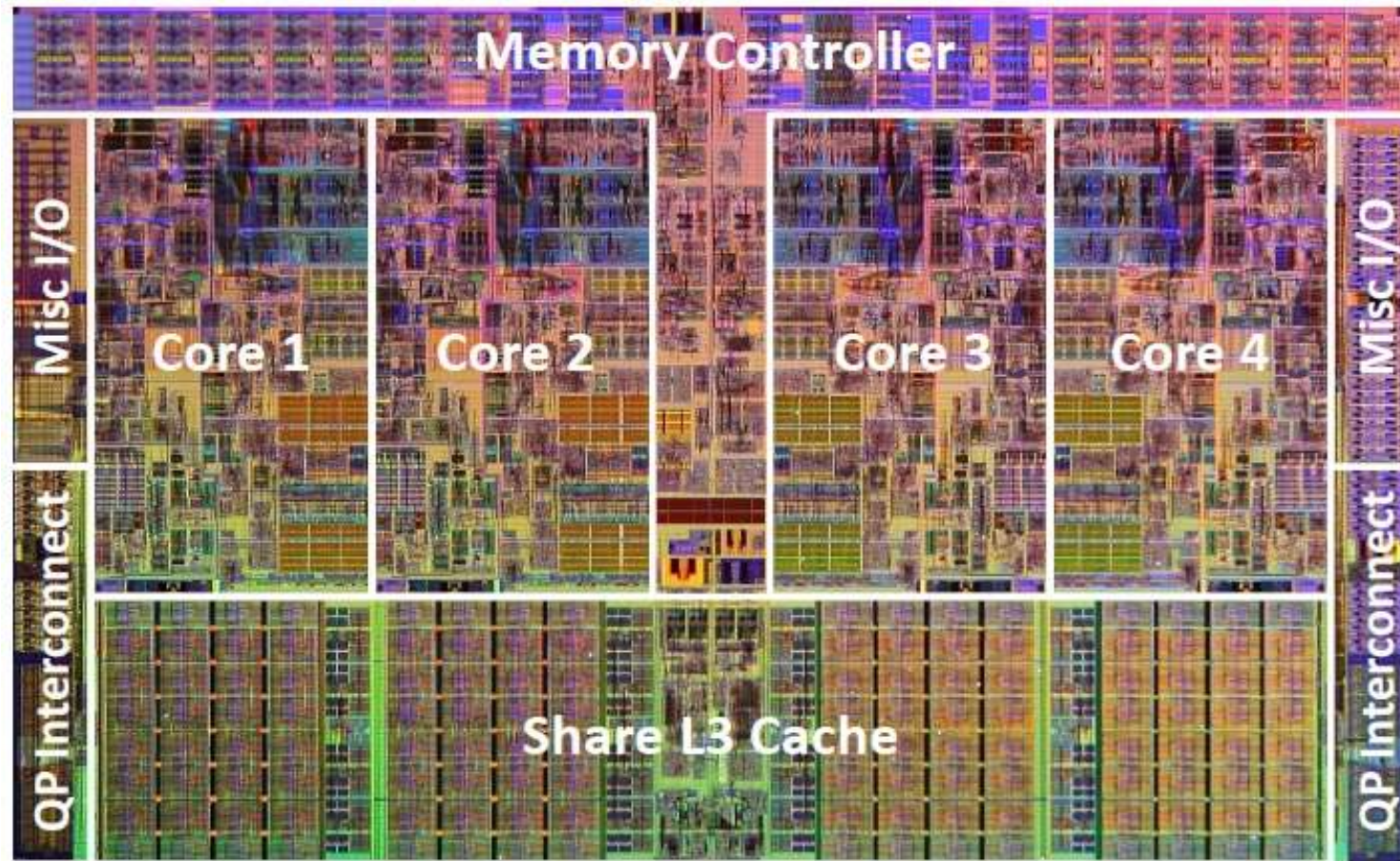
- AMD Barcelona: 4 πυρήνες (cores)



Πηγή: D. Patterson & J. Hennessy: Οργάνωση και Σχεδίαση Υπολογιστών

Συστατικά επεξεργαστή (CPU) (3/3)

- Intel Core i7: 4 πυρήνες (cores)



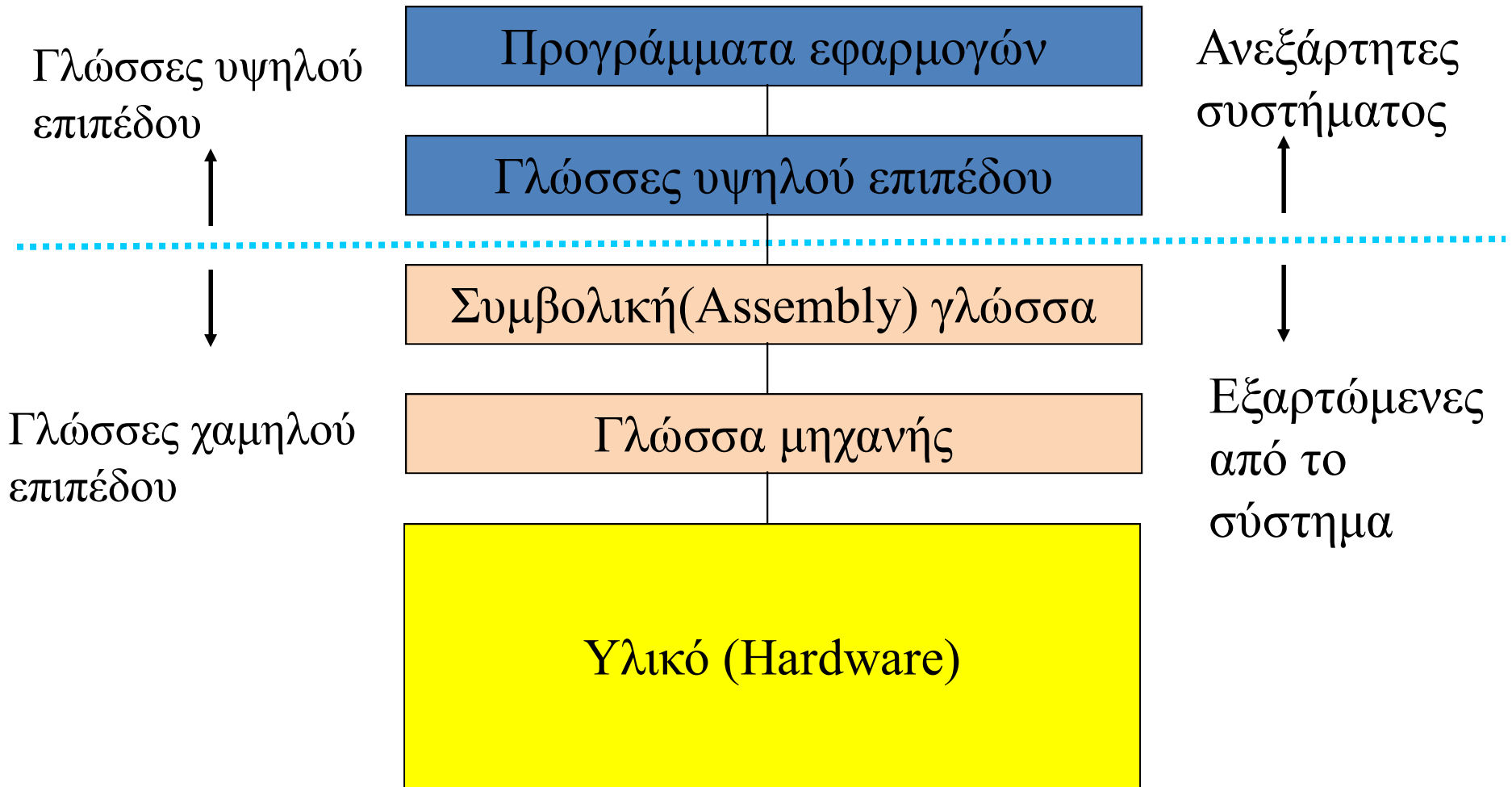
Πηγή: Intel

Κάτω από το πρόγραμμα

- **Λογισμικό εφαρμογών (application Software)**
 - Γραμμένο σε γλώσσα υψηλού επιπέδου (High Level Language, HLL)
- **Λογισμικό συστημάτων (system software)**
 - **Λειτουργικό σύστημα (operating system):**
Κώδικας υπηρεσιών
 - Χειρισμός εισόδου/εξόδου
 - Διαχείριση μνήμης και αποθήκευσης
 - Χρονοπρογραμματισμός εργασιών & κοινή χρήση πόρων
 - **Μεταγλωττιστής (compiler):** Πρόγραμμα που μεταφράζει HLL κώδικα σε κώδικα μηχανής
- **Υλικό (Hardware)**
 - Επεξεργαστής, μνήμη, ελεγκτές εισόδου/εξόδου



Ιεραρχία γλωσσών



Assembly και γλώσσα μηχανής

- Γλώσσα μηχανής
 - Εκτελείται απευθείας από το υλικό (hardware)
 - Οι εντολές αποτελούνται από 0 και 1
- Συμβολική (Assembly) γλώσσα
- Παράδειγμα εντολής γλώσσας μηχανής MIPS:

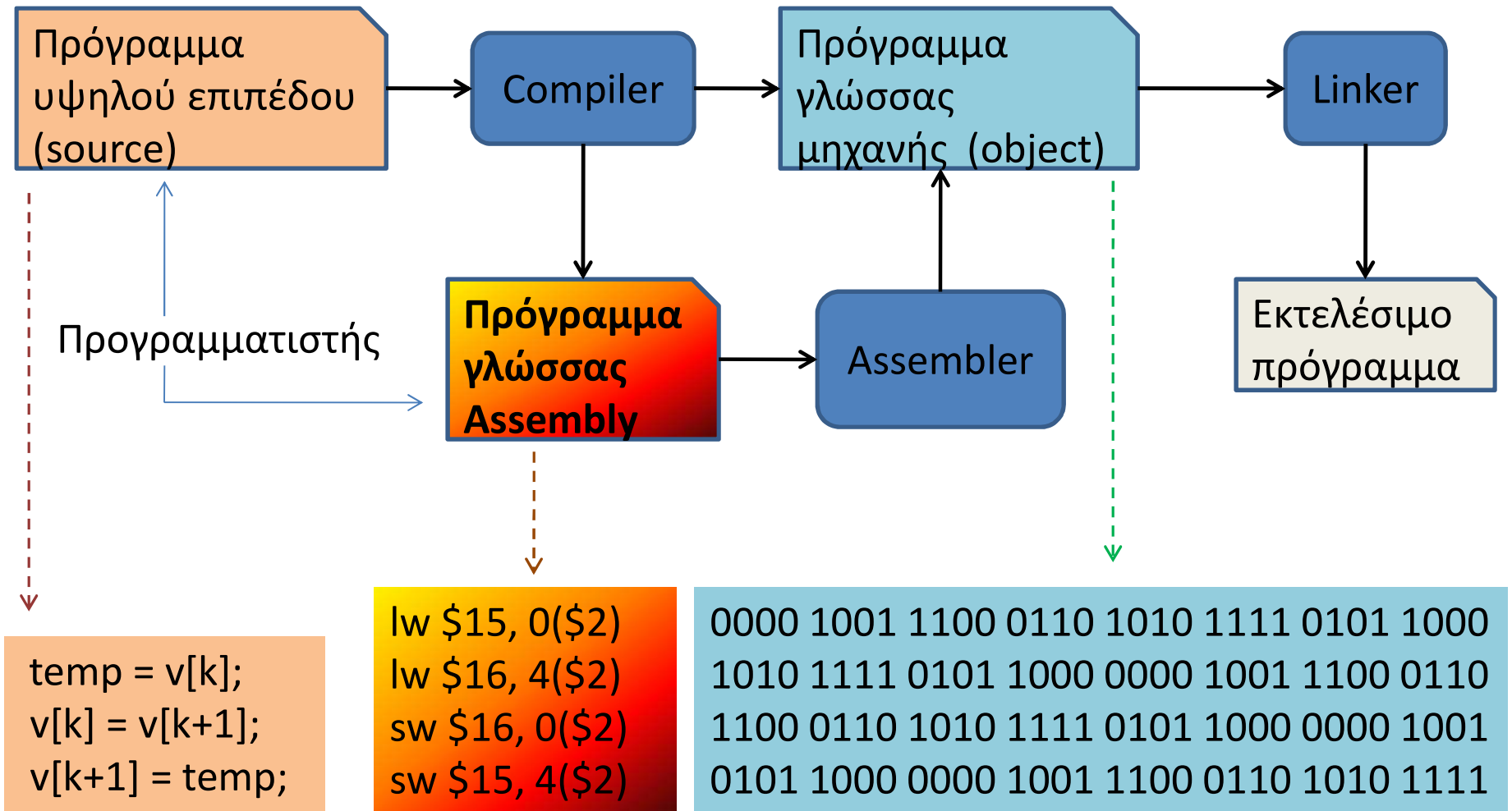
000000 10001 10010 01000 00000 100000

Ισοδύναμη εντολή συμβολικής γλώσσας MIPS:

add \$t0, \$s1, \$s2

\$t0 = \$s1 + \$s2

Compiler - Assembler



Υλικό Υπολογιστή - Κύρια Μνήμη



Main Memory

Addresses	Values
0000000000000000	01111001
0000000000000001	10010100
0000000000000010	10000000
•	•
•	•
•	•
1111111111111101	11110000
1111111111111110	11100000
1111111111111111	00000111

Memory



Υλικό Υπολογιστή - Κεντρική Μονάδα Επεξεργασίας

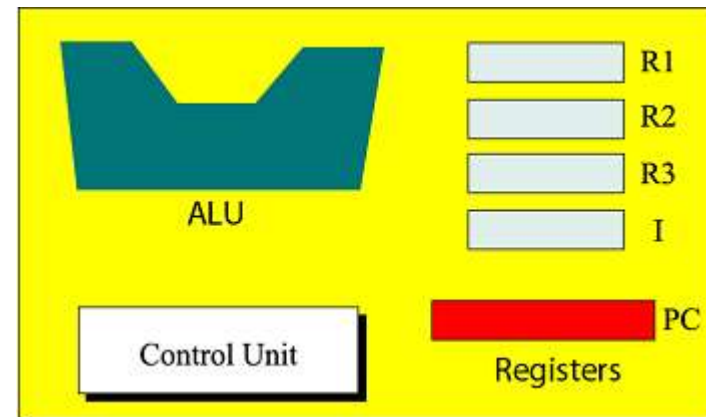
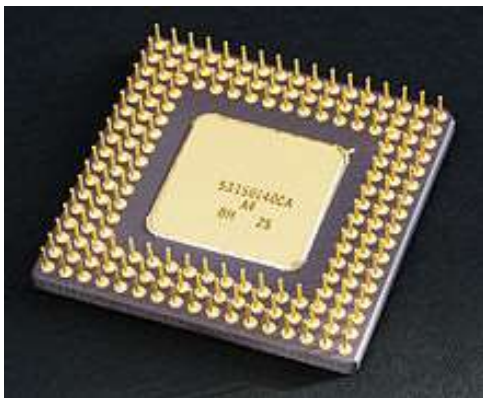
Η είσοδος της ΚΜΕ είναι

- η επόμενη εντολή που πρέπει να εκτελεστεί και
- τα δεδομένα στα οποία πρέπει να εκτελεστεί

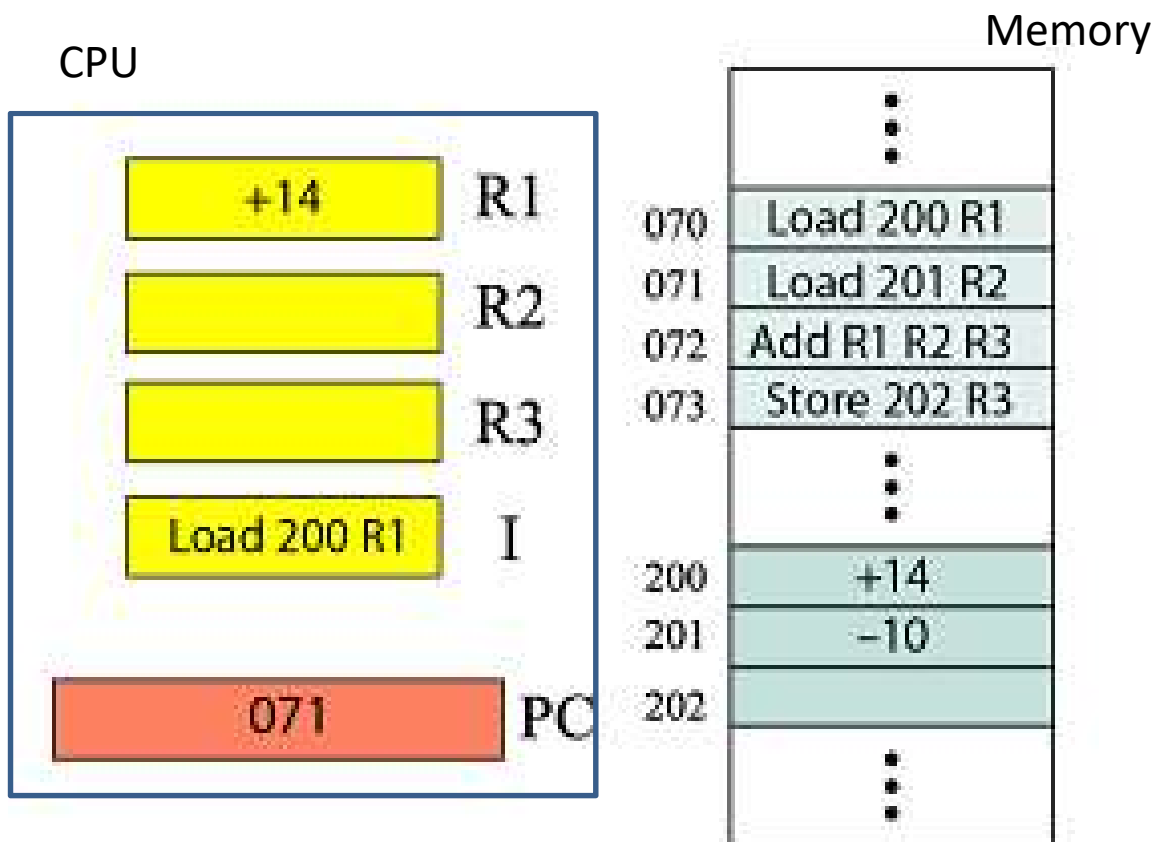
Η έξοδός της είναι τα δεδομένα που προκύπτουν.

Η ΚΜΕ αποτελείται από δύο υπο-μονάδες:

- τη Μονάδα Ελέγχου (ME, Control Unit) και
- την Αριθμητική και Λογική Μονάδα (Arithmetic and Logic Unit, ALU)
- Ένα μικρό πλήθος από καταχωρητές (registers)



Υλικό Υπολογιστή - Κεντρική Μονάδα Επεξεργασίας Και Κύκλος Μηχανής (clock χρονισμού)



Σε ποια στιγμή της εκτέλεσης αντιστοιχεί η εικόνα;
(Ποια εντολή έχει εκτελεστεί);

Μετά το τέλος του προγράμματος Ποια θα είναι τα περιεχόμενα:

- Του R1?
- Του R2?
- Του R3?
- Της θέσης μνήμης 202?

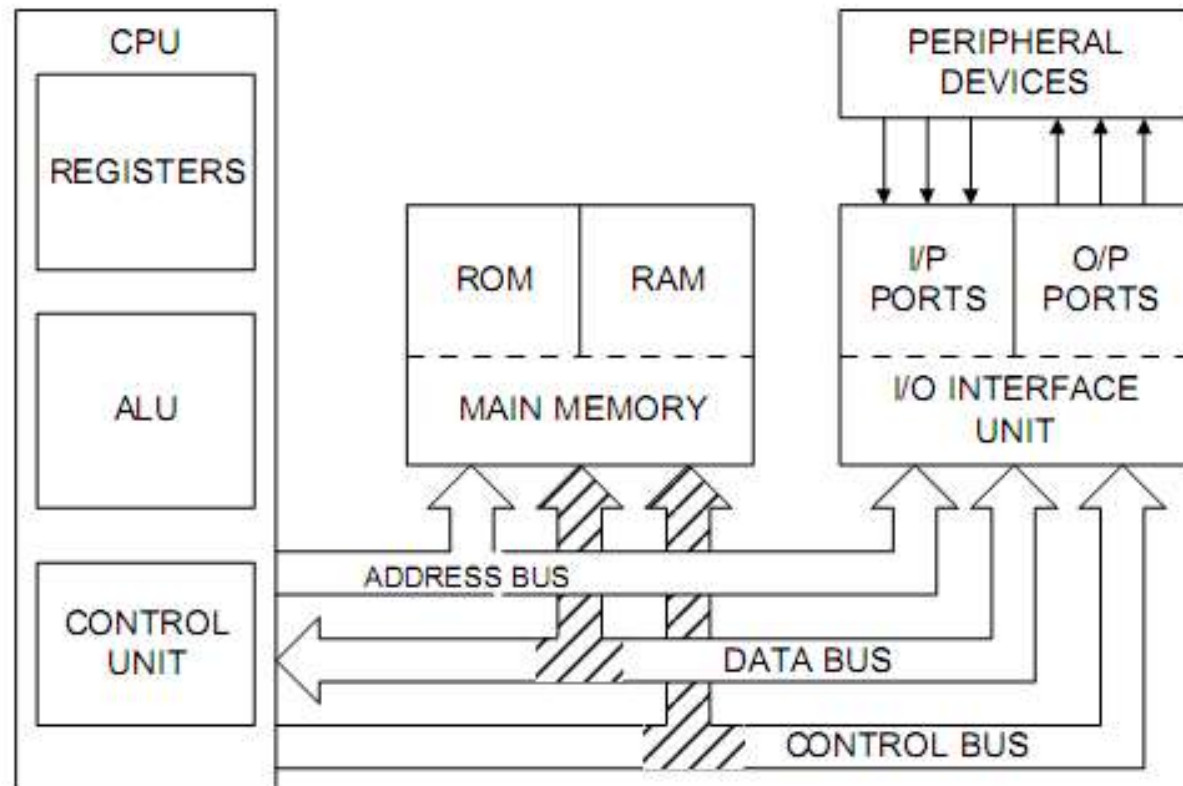
Τι υπάρχει (σε δυαδική μορφή)
Στη θέση 200;
Στη θέση 201;
Στη θέση 70;



Υλικό Υπολογιστή - Κύρια Μνήμη και Επικοινωνία με ΚΜΕ

Η ανταλλαγή μηνυμάτων εκτελείται από τις εξής ομάδες καλωδίων:

- Δίαυλος ελέγχου (control bus),
- Δίαυλος διευθύνσεων (address bus), και
- Δίαυλος δεδομένων (data bus).



Αριθμητικά Συστήματα

Ένας δεκαδικός αριθμός x αποτελείται από μία ακολουθία δεκαδικών ψηφίων και ίσως από μία υποδιαστολή. Οποιοσδήποτε αριθμός (ποσότητα) εκφράζεται ως:

$$(x)_b = \sum_{i=-m}^{n-1} a_i b^i$$

b είναι η βάση του συστήματος ($b \geq 2$)

a_i τα ψηφία του αριθμού αυτού με τιμές 0 έως $b-1$

Ο αριθμός:

$$\alpha_{n-1} b^{n-1} + \alpha_{n-2} b^{n-2} + \dots + \alpha_1 b^1 + \alpha_0 b^0 + \alpha_{-1} b^{-1} + \dots + \alpha_{-m} b^{-m}$$

συμβολίζεται ως: $\alpha_{n-1} \alpha_{n-2} \dots \alpha_1 \alpha_0 \alpha_{-1} \dots \alpha_{-m}$

Παράδειγμα για το δεκαδικό σύστημα: 3347.4

$$= 3 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 + 4 \times 10^{-1}$$

Το σύστημα ονομάζεται **δεκαδικό**, επειδή ως βάση έχει επιλεγεί το 10.

Αριθμητικά Συστήματα

Ως **βάση** ή **ρίζα** ενός αριθμητικού συστήματος ορίζεται το πλήθος των διαφορετικών ψηφίων που χρησιμοποιούνται για την παράσταση των αριθμών

Δεκαδικό σύστημα αρίθμησης

Βάση 10, ψηφία 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Δυαδικό σύστημα αρίθμησης

Βάση 2, ψηφία 0, 1

Οκταδικό σύστημα αρίθμησης

Βάση 8, ψηφία 0, 1, 2, 3, 4, 5, 6, 7

Δεκαεξαδικό σύστημα αρίθμησης

Βάση 16, ψηφία 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Μετατροπή

από οποιοδήποτε σύστημα στο δεκαδικό

- Ο τύπος της γενικής αναπαράστασης ενός αριθμού σε οποιοδήποτε σύστημα μας δίνει και την αξία του στο δεκαδικό σύστημα αρίθμησης (αφού γίνουν οι πράξεις).
- Πολλαπλασιάζω **κάθε ψηφίο** του αριθμού που δίνεται με τη **βάση του ίδιου του συστήματος** υψωμένη σε **δύναμη** που προσδιορίζεται από τη θέση του ψηφίου στον αριθμό.
- Στο **ακέραιο μέρος** ξεκινώ από το τελευταίο ψηφίο του (μονάδες) το οποίο βρίσκεται στη θέση 0 και συνεχίζω προς την αρχή του (κινούμαι αριστερά) αυξάνοντας συνεχώς τον αριθμό της θέσης κατά ένα.
- Στο **κλασματικό μέρος** κινούμαι δεξιά θεωρώντας ότι το πρώτο κλασματικό ψηφίο βρίσκεται στη θέση -1 και σε κάθε βήμα μειώνω κατά ένα τον αριθμό της θέσης (-2, -3 κ.ο.κ.).

$$A^1=A, A^0=1, A^{-1}=1/A, A^{-2}=1/A^2$$

Παραδείγματα

$$(1673,42)_{10} =$$

$$1 \times 10^3 + 6 \times 10^2 + 7 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 2 \times 10^{-2}$$

$$(100110)_2 =$$

$$1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (38)_{10}$$

$$(372)_8 =$$

$$3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = (250)_{10}$$

$$(A34F,4)_{16} =$$

$$A \times 16^3 + 3 \times 16^2 + 4 \times 16^1 + F \times 16^0 + 4 \times 16^{-1} = (41807,25)_{10}$$

Μετατροπές

Από το δεκαδικό σύστημα σε οποιοδήποτε άλλο (1)

- Έστω αριθμός με **ακέραιο και κλασματικό μέρος** ($γγγγ,χχχ$).
- Χωρίζω το ακέραιο από το κλασματικό μέρος $γγγγ$ και $0,χχχ$
- Το **ακέραιο μέρος** του αριθμού διαιρείται (ακέραια διαίρεση) συνεχώς με τη βάση του συστήματος στο οποίο θέλουμε να το μετατρέψουμε.
- Η διαίρεση σταματάει όταν το πηλίκο γίνει 0.
- Ο ζητούμενος αριθμός προκύπτει παίρνοντας ανάποδα όλα τα υπόλοιπα των διαιρέσεων.

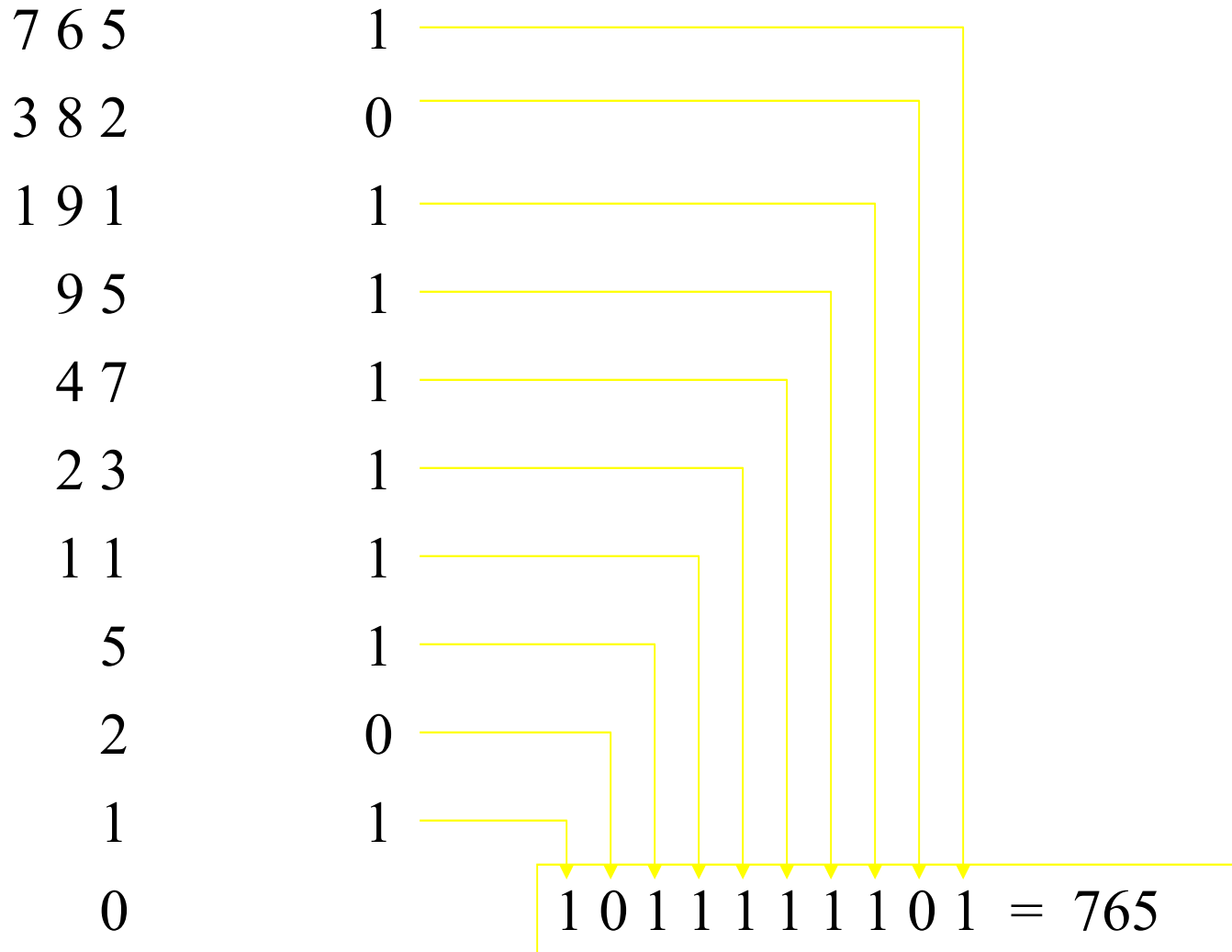
Μετατροπές

Από το δεκαδικό σύστημα σε οποιοδήποτε άλλο (2)

- Το κλασματικό μέρος πολλαπλασιάζεται συνεχώς με τη βάση του συστήματος στο οποίο θέλουμε να το μετατρέψουμε.
- Οποιοδήποτε ακέραιο μέρος παραχθεί κατά τον πολλαπλασιασμό, διαχωρίζεται πάλι από το κλασματικό μέρος (με βάση την παραπάνω λογική) και η διαδικασία επαναλαμβάνεται.
- Ο πολλαπλασιασμός σταματάει όταν φτάσουμε το πλήθος των δεκαδικών ψηφίων που θέλουμε ή όταν βρούμε κλασματικό μέρος 0.
- Ο κλασματικός αριθμός στο νέο σύστημα σχηματίζεται παίρνοντας όλα τα ακέραια μέρη όλων των πολλαπλασιασμών που κάναμε με φορά από τον πρώτο προς τον τελευταίο.

Μετατροπές (ακέραιο μέρος)

Από δεκαδικό σε δυαδικό του 765 (Διαδοχικές διαιρέσεις με το 2):



Μετατροπές (δεκαδικό μέρος)

Από δεκαδικό σε δυαδικό του 0,41

(Διαδοχικοί πολλαπλασιασμοί με το 2):

0,41	2	0,82	0
0,82	2	1,64	1
0,64	2	1,28	1
0,28	2	0,56	0
0,56	2	1,12	1

= 0,01101

Μετατροπές (δεκαδικό μέρος)

Από δεκαδικό σε δεκαεξαδικό του 0,23

(Διαδοχικοί πολλαπλασιασμοί με το 16):

0,23	16	3,68	3
0,68	16	10,88	A
0,88	16	14,08	E
0,08	16	1,28	1
0,28	16	4,48	4

= 0,3AE14

Μετατροπές (ακέραιο μέρος)

Από δεκαδικό σε οκταδικό του 7653 (Διαδοχικές διαιρέσεις με το 8):

7 6 5 3

9 5 6

1 1 9

1 4

1

5

4

7

6

1

(1 6 7 4 5)₈

Μετατροπές - Παράγωγα συστήματα

Συστήματα που η βάση του ενός είναι η ύψωση σε δύναμη της βάσης ενός άλλου ονομάζονται **παράγωγα συστήματα αρίθμησης**.

Παράδειγμα 1: το εννεαδικό σύστημα αρίθμησης είναι παράγωγο του τριαδικού γιατί $3^2=9$.

Παράδειγμα 2: το δεκαεξαδικό σύστημα αρίθμησης είναι παράγωγο του δυαδικού γιατί $2^4=16$.

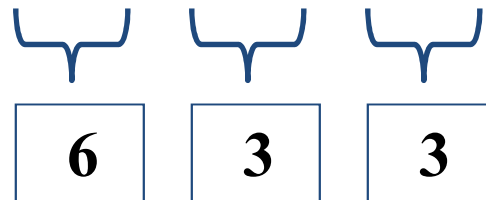
Κάθε ψηφίο του **δεκαεξαδικού αριθμού** μετατρέπεται αυτόνομα στο **δυαδικό σύστημα** (χρησιμοποιώντας 4 δυαδικά ψηφία) και έτσι προκύπτει ο δυαδικός αριθμός και αντίστροφα.

Αντίστοιχα, κάθε ψηφίο του **οκταδικού αριθμού** μετατρέπεται αυτόνομα στο **δυαδικό σύστημα** (χρησιμοποιώντας 3 δυαδικά) και έτσι προκύπτει ο δυαδικός αριθμός και αντίστροφα.

Μετατροπές - Παράγωγα συστήματα

Δυαδικό		Οκταδικό		Δυαδικό		Οκταδικό
0 0 0	↔	0		1 0 0	↔	4
0 0 1	↔	1		1 0 1	↔	5
0 1 0	↔	2		1 1 0	↔	6
0 1 1	↔	3		1 1 1	↔	7

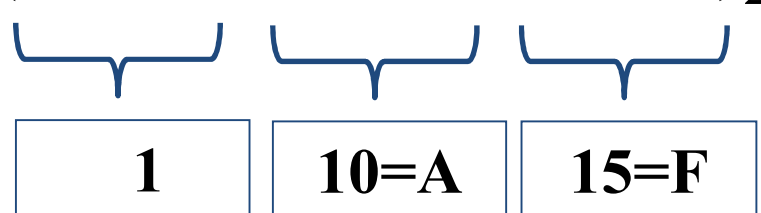
$$(633)_8 = (1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1)_2$$



Μετατροπές-Παράγωγα συστήματα

Δυαδικό	Δεκαεξαδικό	Δυαδικό	Δεκαεξαδικό
0 0 0 0	↔ 0	1 0 0 0	↔ 8
0 0 0 1	↔ 1	1 0 0 1	↔ 9
0 0 1 0	↔ 2	1 0 1 0	↔ A
0 0 1 1	↔ 3	1 0 1 1	↔ B
0 1 0 0	↔ 4	1 1 0 0	↔ C
0 1 0 1	↔ 5	1 1 0 1	↔ D
0 1 1 0	↔ 6	1 1 1 0	↔ E
0 1 1 1	↔ 7	1 1 1 1	↔ F

$$(1AF)_{16} = (0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1)_2$$



1	10=A	15=F
---	------	------

Μετατροπές - Παράγωγα συστήματα

Από δυαδικό σε οκταδικό

$$(1\ 0\ 1\ 0\ 1\ 1\ 1,0\ 1\ 1\ 0\ 1)_2$$

Χωρίζουμε σε τριάδες από την υποδιαστολή προς τα αριστερά (ακέραιο μέρος) και από την υποδιαστολή προς τα δεξιά (δεκαδικό μέρος)

(κατά περίπτωση προσθέτουμε μηδενικά)

$$(1\ 0\ 1\ 0\ 1\ 1\ 1,0\ 1\ 1\ 0\ 1)_2$$

1	2	7	3	2
---	---	---	---	---

Μετατροπές - Παράγωγα συστήματα

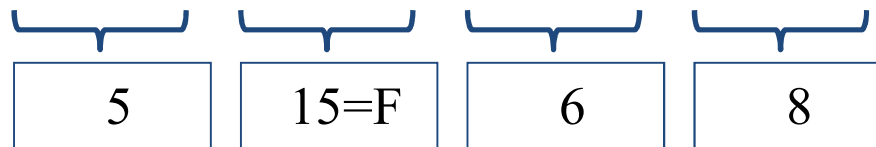
Από δυαδικό σε δεκαεξαδικό

$$(1\ 0\ 1\ 1\ 1\ 1\ 1,0\ 1\ 1\ 0\ 1)_2$$

Χωρίζουμε σε τετράδες από υποδιαστολή προς τα αριστερά (ακέραιο μέρος) και από υποδιαστολή προς τα δεξιά (δεκαδικό μέρος)

(κατά περίπτωση προσθέτουμε μηδενικά)

$$(1\ 0\ 1\ 1\ 1\ 1\ 1,0\ 1\ 1\ 0\ 1)_2$$



Πράξεις (πρόσθεση)

Η πρόσθεση ανεξάρτητα από το αριθμητικό σύστημα γίνεται θεωρώντας το κάθε ζεύγος (μονοψήφιων) αριθμών που προστίθεται ως δεκαδικό

αλλά

το αποτέλεσμα της πράξης γράφεται πάντα στο σύστημα στο οποίο γίνεται η πράξη (με τη λογική του αποτελέσματος και του κρατουμένου που ισχύει και στο δεκαδικό σύστημα).

Πρόσθεση στο δυαδικό σύστημα

Πρόσθεση δυαδικών:

A	+	B	Κρατούμενο	Αποτέλεσμα
0		0	0	0
0		1	0	1
1		0	0	1
1		1	1	0

Πρόσθεση στο δυαδικό σύστημα

Κρατούμενο Εισόδου	A	B	Κρατούμενο εξόδου	Αποτέλεσμα
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Παράδειγμα:

1ος Προσθετέος:	0	0	1	1	1	$(7)_{10}$
2ος Προσθετέος:	0	1	0	1	0	$(10)_{10}$
Άθροισμα:	1	0	0	0	1	$(17)_{10}$
Κρατούμενο:	0	1	1	1	0	

Παράσταση αρνητικών αριθμών: Παράσταση μέτρου

Λαμβάνεται αλλάζοντας το υψηλής τάξης ψηφίο από μηδέν σε ένα

0 0 0 1 0 0 1 1

1 0 0 1 0 0 1 1

Το πιο σημαντικό bit (MSB) χρησιμοποιείται για τον προσδιορισμό του προσήμου.

Γενικά αναπαρίστανται οι $[-(2^{n-1}-1), +(2^{n-1}-1)]$

Θετικοί		Αρνητικοί	
0000	0	1000	-0
0001	+1	1001	-1
0010	+2	1010	-2
0011	+3	1011	-3
0100	+4	1100	-4
0101	+5	1101	-5
0110	+6	1110	-6
0111	+7	1111	-7

Παράσταση συμπληρώματος ως προς ένα

Το **συμπλήρωμα ως προς ένα** μιας δυαδικής ακολουθίας λαμβάνεται αλλάζοντας όλα τα bit από μηδέν σε ένα και αντιστρόφως

1 0 0 1 0 0 1 1

0 1 1 0 1 1 0 0

Το πιο σημαντικό bit (MSB) χρησιμοποιείται για τον προσδιορισμό του προσήμου.

Γενικά αναπαριστώνται οι $[-(2^{n-1}-1), +(2^{n-1}-1)]$

Θετικοί		Αρνητικοί	
0000	0	1000	-7
0001	+1	1001	-6
0010	+2	1010	-5
0011	+3	1011	-4
0100	+4	1100	-3
0101	+5	1101	-2
0110	+6	1110	-1
0111	+7	1111	-0

Παράσταση συμπληρώματος ως προς δύο

Το συμπλήρωμα ως προς δύο μιας δυαδικής ακολουθίας λαμβάνεται αλλάζοντας όλα τα bit από μηδέν σε ένα και αντιστρόφως και στη συνέχεια προσθέτοντας το '1'

1 0 0 1 0 0 1 1

0 1 1 0 1 1 0 0

0 1 1 0 1 1 0 1

Το MSB χρησιμοποιείται **και εδώ** για τον προσδιορισμό του προσήμου.

Στη γενική περίπτωση αναπαριστώνται οι $[-2^{n-1}, +(2^{n-1}-1)]$

Θετικοί + 0		Αρνητικοί	
0000	0	1000	-8
0001	+1	1001	-7
0010	+2	1010	-6
0011	+3	1011	-5
0100	+4	1100	-4
0101	+5	1101	-3
0110	+6	1110	-2
0111	+7	1111	-1

Παράδειγμα

Προσοχή: θεωρούμε συγκεκριμένο αριθμό bits

για $n=6$ bits, $X_{10} = -17 \Rightarrow$

Εύρεση συμπληρώματος ως προς 2 του αριθμού 17 :

$X = 17$	\rightarrow	010001	
		101110	(Αντιστροφή όλων των bit)
		101111	(Πρόσθεση του 1)

Πρόσθεση με Αριθμητική Συμπληρώματος του 2

Το αλγεβρικό άθροισμα δύο αριθμών στην παράσταση συμπληρώματος του 2 προκύπτει ως το δυαδικό άθροισμα των δύο αριθμών, αγνοώντας το τυχόν κρατούμενο:

+12	001100	+12	001100
+17	010001	- 17	101111
-----	-----	-----	-----
29	011101	- 5	111011

Πρόσθεση με Αριθμητική Συμπληρώματος του 2

Το αλγεβρικό άθροισμα δύο αριθμών στην παράσταση συμπληρώματος του 2 προκύπτει ως το δυαδικό άθροισμα των δύο αριθμών, αγνοώντας το τυχόν κρατούμενο:

$$\begin{array}{r} -12 \quad 110100 \\ +17 \quad 010001 \\ \hline +5 \quad 000101 \end{array}$$

**Αγνοείται το
κρατούμενο 1**

$$\begin{array}{r} -12 \quad 110100 \\ -17 \quad 101111 \\ \hline -29 \quad 100011 \end{array}$$

**Αγνοείται το
κρατούμενο 1**

Αριθμητική Υπολογιστών

Αριθμητική Πεπερασμένης Ακρίβειας: Καθώς η ποσότητα της διαθέσιμης μνήμης για την αποθήκευση ενός αριθμού είναι καθορισμένη, οι αριθμοί μπορούν να αναπαρασταθούν με έναν καθορισμένο αριθμό ψηφίων.

Έστω ότι για την αναπαράσταση θετικών ακεραίων διατίθενται **τρία ψηφία του δεκαδικού συστήματος.**

Με αυτό τον περιορισμό **δεν** μπορούμε να εκφράσουμε:

- Αριθμούς μεγαλύτερους από 999
- Αρνητικούς αριθμούς
- Κλάσματα
- Μιγαδικούς αριθμούς

Πεπερασμένη ακρίβεια

Οι αριθμοί πεπερασμένης ακρίβειας δεν είναι κλειστοί ως προς τις πράξεις :

$$600 + 600 = 1200 \quad (\text{πολύ μεγάλος})$$

$$003 - 005 = -2 \quad (\text{αρνητικός})$$

$$050 \times 050 = 2500 \quad (\text{πολύ μεγάλος})$$

$$007 / 002 = 3.5 \quad (\text{όχι ακέραιος})$$

Μπορεί να έχουμε:

- **σφάλμα υπερχείλισης (overflow)**, δηλαδή το αποτέλεσμα είναι μεγαλύτερο από τον μεγαλύτερο αριθμό του συνόλου,
- **σφάλμα ανεπάρκειας (underflow)**, όταν το αποτέλεσμα είναι μικρότερο από τον μικρότερο αριθμό του συνόλου.

Αναπαράσταση αριθμών και σφάλματα

- ❑ Ο περιορισμός στο μέγεθος της λέξης της μνήμης έχει ως συνέπεια να μπορούν να αποθηκευθούν μόνο στρογγυλοποιημένες, προσεγγιστικές τιμές πραγματικών αριθμών.
- ❑ Επομένως οι υπολογιστές στην πραγματικότητα κάνουν λάθη σε κάθε έναν από τους υπολογισμούς που πραγματοποιούν.
 - ❑ Μικρά, αλλά πολλά.
- ❑ Όλα αυτά τα μικρά λάθη μπορούν να συσσωρευτούν και να αυξηθούν μέσα σε κάθε υπολογισμό που γίνεται και τελικά να δώσουν ένα αποτέλεσμα που είναι αρκετά διαφορετικό από το ακριβές.

Υπολογισμοί και σφάλματα (1)

- Στις 25/2/1991, στην πόλη Νταχράν της Σ. Αραβίας, κατά την διάρκεια του πόλεμου στον κόλπο, μία Αμερικάνικη συστοιχία πυραύλων τύπου Patriot, απέτυχε να εντοπίσει και να εξουδετερώσει έναν Ιρακινό πύραυλο τύπου Scud.
- Ο Scud χτύπησε ένα στόχο του Αμερικάνικου στρατού, σκοτώνοντας 28 στρατιώτες και τραυματίζοντας 100.
- Αίτιο της αποτυχίας ήταν τα λάθη στρογγυλοποιήσεων στον υπολογισμό του χρόνου.
- Ειδικότερα: ο χρόνος χωριζόταν σε δέκατα του second και χρησιμοποιούσε ένα καταχωρητή 24 δυαδικών θέσεων για αποθήκευση.

Αριθμητική Υπολογιστών

- Υπολογιστής με μήκος λέξης τέσσερα ψηφία

Μόνο θετικοί Αριθμοί		Αρνητικοί - Παράσταση μέτρου		Αρνητικοί - ΠΣ2	
0000	0	0000	0	0000	0
0001	1	0001	1	0001	1
0010	2	0010	2	0010	2
0011	3	0011	3	0011	3
0100	4	0100	4	0100	4
0101	5	0101	5	0101	5
0110	6	0110	6	0110	6
0111	7	0111	7	0111	7
1000	8	1000	-0	1000	-8
1001	9	1001	-1	1001	-7
1010	10	1010	-2	1010	-6
1011	11	1011	-3	1011	-5
1100	12	1100	-4	1100	-4
1101	13	1101	-5	1101	-3
1110	14	1110	-6	1110	-2
1111	15	1111	-7	1111	-1

Αριθμητική Υπολογιστών

- Μπορώ να αναπαραστήσω κλασματικούς αριθμούς;
- Αριθμητική σταθερής υποδιαστολής
- Σύμβαση για τη θέση της υποδιαστολής

Αριθμητική Υπολογιστών

- Υπολογιστής με μήκος λέξης τέσσερα ψηφία, ένα για υποδιαστολή (θετικοί)

0000	000,0	0
0001	000,1	0,5
0010	001,0	1
0011	001,1	1,5
0100	010,0	2
0101	010,1	2,5
0110	011,0	3
0111	011,1	3,5
1000	100,0	4
1001	100,1	4,5
1010	101,0	5
1011	101,1	5,5
1100	110,0	6
1101	110,1	6,5
1110	111,0	7
1111	111,1	7,5

Αριθμητική Υπολογιστών

- Υπολογιστής με μήκος λέξης τέσσερα ψηφία, ένα για υποδιαστολή, ΠΣ2

0000	000,0	0
0001	000,1	0,5
0010	001,0	1
0011	001,1	1,5
0100	010,0	2
0101	010,1	2,5
0110	011,0	3
0111	011,1	3,5
1000	100,0	-4
1001	100,1	-3,5
1010	101,0	-3
1011	101,1	-2,5
1100	110,0	-2
1101	110,1	-1,5
1110	111,0	-1
1111	111,1	-0,5

Αριθμητική Υπολογιστών

- Μπορώ να παραστήσω πιο μεγάλους ή πιο μικρούς αριθμούς;
- Αριθμητική κινητής υποδιαστολής
- Θέματα για προβληματισμό:
 - Τι γίνεται για πολύ μεγάλους αριθμούς;
 - Τι γίνεται για αποτελέσματα της μορφής 0/0?

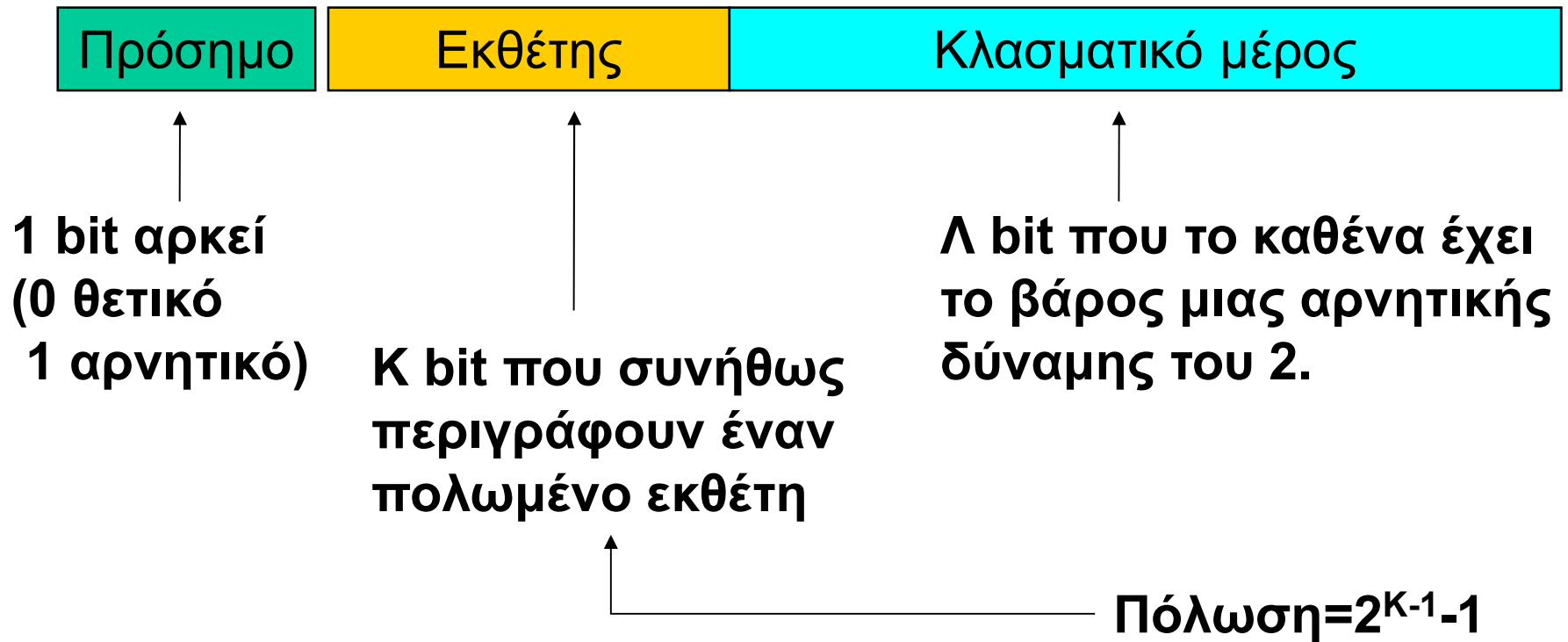
Αριθμητική Υπολογιστών

- Κινητή υποδιαστολή

$$x = \textit{sign} \times \textit{fraction} \times \textit{base}^{\textit{exponent} - \textit{bias}}$$

- **sign** το πρόσημο του αριθμού
- **fraction (mantissa, magnitude, significant, σημαντικό μέρος)**
κανονικοποιείται έτσι ώστε το πρώτο bit να είναι πάντα 1.
- Αυτό το bit δεν αποθηκεύεται (για λόγους οικονομίας) και ονομάζεται κρυφό bit.
- **base (radix)** η βάση του αριθμητικού συστήματος στο οποίο αναπαρίσταται ο αριθμός (2 για το δυαδικό 10 για το δεκαδικό)
- **exponent (characteristic)** είναι ο εκθέτης της βάσης
- **bias (πόλωση)** είναι ένας ακέραιος αριθμός

Παράσταση κινητής υποδιαστολής



Στο IEEE 754 έχουμε 2 τέτοιες αναπαραστάσεις

- A) Απλής ακρίβειας (K=8, Λ=23, σύνολο 32 bit, Πόλωση=127) και
- B) Διπλής ακρίβειας (K=11, Λ=52, σύνολο 64 bit, Πόλωση=1023)

Πρότυπο IEEE 754

S Πρόσημο

1 bit

E Εκθέτης

8 bits

Σ Συντελεστής

23 bits

αριθμός $X = (-1)^S \cdot (1 + \Sigma) \cdot 2^{E-127}$,

πόλωση = 127

(s=πρόσημο, «0»=θετικός, «1»=αρνητικός)

Πρότυπο ΙΕΕΕ 754 – Παράδειγμα 1

S	E Εκθέτης	Σ Συντελεστής
0	1000 0010	1100 0000 0000 0000 0000 000

Ποιος αριθμός είναι αυτός;

εκθέτης (πολωμένος) $E=1000\ 0010 = 130$

$\Sigma=1100\ 0000\ 0000\ 0000\ 0000\ 000 = 0.75$

$$\alpha = (-1)^S \cdot (1 + \Sigma) \cdot 2^{E-127}$$

$$\alpha = (-1)^0 \cdot (1 + 0.75) \cdot 2^{130-127}$$

$$\alpha = + (1.75) \cdot 2^3$$

$$\alpha = +14$$

IEEE 754 – Παράδειγμα 2

S E Εκθέτης

Σ Συντελεστής

0 0111 1110

1000 0000 0000 0000 0000 000

εκθέτης (πολωμένος) $E=0111\ 1110=126$, $\Sigma=0.5$

$$\alpha = (-1)^0 \cdot (1+0.5) \cdot 2^{126-127} = +(1.5) \cdot 2^{-1} = +0.75$$

S E Εκθέτης

Σ Συντελεστής

0 1000 0000

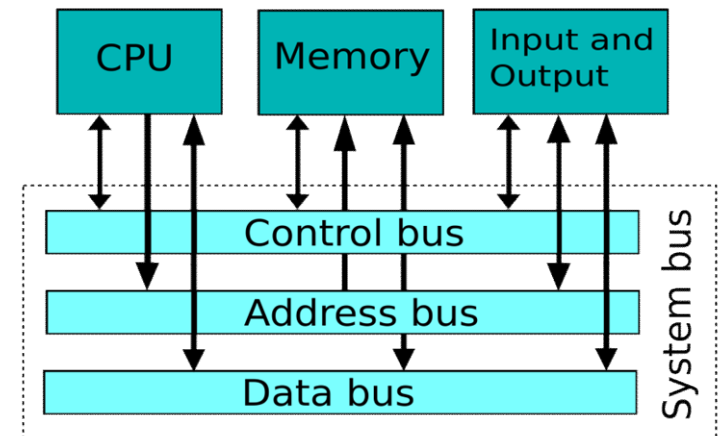
1000 0000 0000 0000 0000 000

εκθέτης (πολωμένος) $E=1000\ 0000=128$, $\Sigma=0.5$

$$\alpha = (-1)^0 \cdot (1+0.5) \cdot 2^{128-127} = +(1.5) \cdot 2^{+1} = +3$$

Αρχιτεκτονική von Neumann

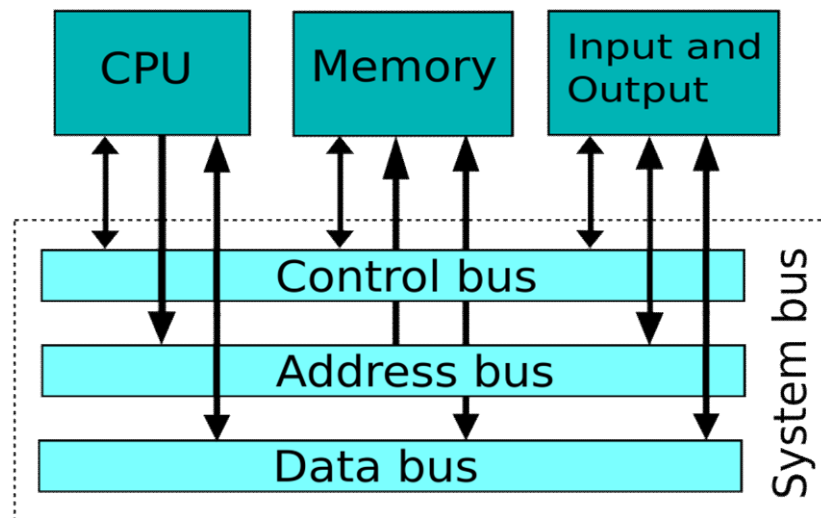
- Οι σύγχρονοι ηλεκτρονικοί υπολογιστές σχεδιάζονται με βάση τις αρχές που διατυπώθηκαν από τον John von Neumann στο Ινστιτούτο Προηγμένων Επιστημών στο Πανεπιστήμιο του Πρίνστον το 1945.
- Στις αρχές αυτές αναφερόμαστε με τον όρο αρχιτεκτονική von Neumann (μοντέλο von Neumann) ή αρχιτεκτονική Πρίνστον.
- Η αρχιτεκτονική von Neumann προβλέπει την ύπαρξη ενός επεξεργαστή, ο οποίος επικοινωνεί με τη μνήμη καθώς και με μια σειρά από συσκευές εισόδου/ εξόδου.



Δίαυλοι

Η επικοινωνία μεταξύ των διαφόρων τμημάτων του υπολογιστή εκτελείται από ομάδες καλωδίων που ονομάζονται δίαυλοι (buses):

- Δίαυλος ελέγχου (control bus)
- Δίαυλος διευθύνσεων (address bus)
- Δίαυλος δεδομένων (data bus)

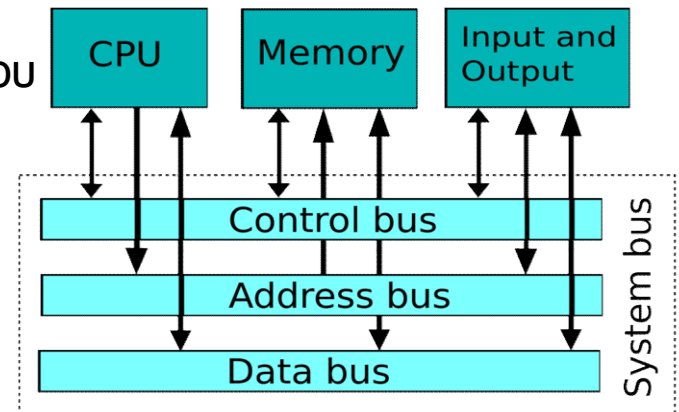


Δίαυλοι - Άσκηση (1)

Έστω ένας υπολογιστής με τα ακόλουθα χαρακτηριστικά:

- Address bus: 8 bits
- Data bus: 8 bits
- Control bus: 4 bits
 - Read from mem
 - Write to mem
 - Read from I/O
 - Write to I/O
- Πληκτρολόγιο (Διεύθυνση 0x00)
- Οθόνη (Διεύθυνση 0x01)
- Οι πρώτες θέσεις της μνήμης έχουν τα περιεχόμενα που φαίνονται στο διπλανό πίνακα

Διεύθυνση	Περιεχόμενο
0x00	0xFF
0x01	0xAC
0x02	0x32



Δίαυλοι - Άσκηση (2)

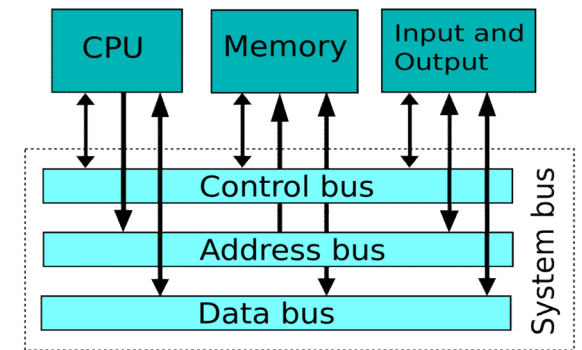
Έστω οι ακόλουθες ενέργειες που θέλει να εκτελέσει ο επεξεργαστής.

- Εντολή 1: Θέλω να διαβάσω το περιεχόμενο της μνήμης με διεύθυνση 0x01.
- Εντολή 2: Θέλω να γράψω στη θέση μνήμης με διεύθυνση 0x02 το περιεχόμενο 0xFF.

Τι πληροφορία θα έχουν οι δίαυλοι;

Διεύθυνση	Περιεχόμενο
0x00	0xFF
0x01	0xAC
0x02	0x32

	Read from mem	Write to mem	Read from I/O	Write to I/O	Address bus	Data bus
Εντολή 1						
Εντολή 2						



	Read from mem	Write to mem	Read from I/O	Write to I/O	Address bus	Data bus
Εντολή 1	1	0	0	0	00000001	10101100
Εντολή 2	0	1	0	0	00000010	11111111

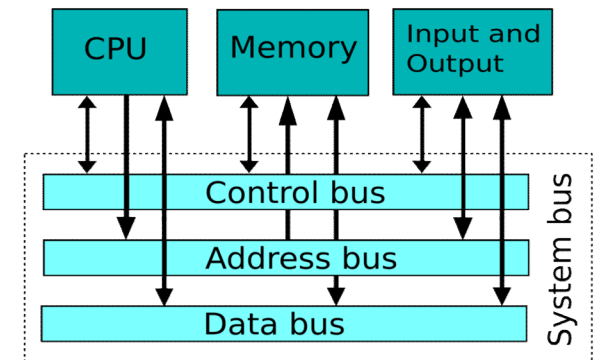
Δίαυλοι – Άσκηση (3)

Έστω οι ακόλουθες ενέργειες που θέλει να εκτελέσει ο επεξεργαστής.

- Εντολή 3: Θέλω να διαβάσω από το πληκτρολόγιο (διεύθυνση 0x00) το χαρακτήρα που πατήθηκε (έστω το 'a' (ascii 0x97))
- Εντολή 4: Θέλω να γράψω στην οθόνη (διεύθυνση 0x01) το χαρακτήρα 0x20

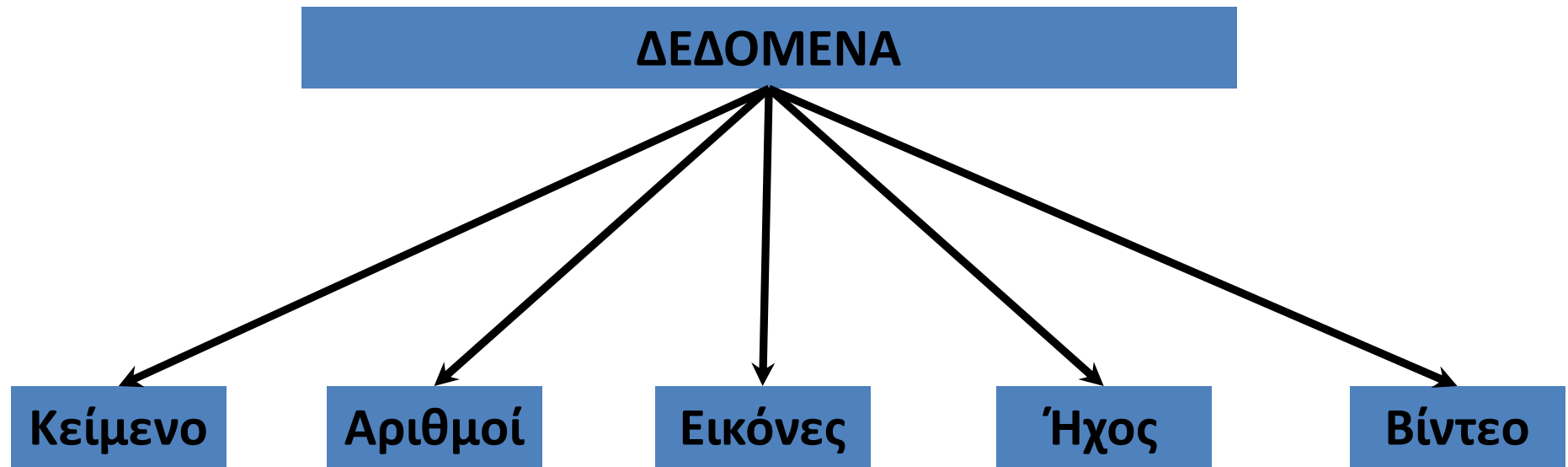
Τι πληροφορία θα υπάρχει σε κάθε ένα από τους διαύλους;

	Read from mem	Write to mem	Read from I/O	Write to I/O	Address bus	Data bus
Εντολή 3						
Εντολή 4						



	Read from mem	Write to mem	Read from I/O	Write to I/O	Address bus	Data bus
Εντολή 3	0	0	1	0	00000000	10010111
Εντολή 4	0	0	0	1	00000001	00100000

Αναπαράσταση της πληροφορίας στον Υπολογιστή



Bits και Bytes

- Το **bit** (μπιτ) (συμβολίζεται ως **b**) είναι ένα δυαδικό ψηφίο το οποίο μπορεί να πάρει τις τιμές 0 ή 1.
- Οι υπολογιστές εργάζονται με το **δυαδικό σύστημα αρίθμησης** και χρησιμοποιούν δυαδικά ψηφία για να συμβολίζουν **εντολές** και **δεδομένα**.
- Η συντομογραφία bit προέρχεται από τη σύντμηση των λέξεων **BI**nary digiT.



Bits και Bytes

Το **byte** είναι **μονάδα μέτρησης** ποσότητας πληροφορίας στα υπολογιστικά συστήματα.

1 byte ισοδυναμεί με 8 bit.

Το byte μπορεί να αντιπροσωπεύσει τιμές από 0 έως και 255 στο δεκαδικό σύστημα ($2^8 = 256$ τιμές).

Το byte είναι και η βασική μονάδα μέτρησης (χώρου και πληροφορίας) στα υπολογιστικά συστήματα. Πολλαπλάσιά του είναι τα:

μονάδα	πλήθος μπάιτ	προσέγγιση
1 kilobyte (KB)	$2^{10}=1.024$	10^3
1 megabyte (MB)	$2^{20}=1.048.576$	10^6
1 gigabyte (GB)	$2^{30}=1.073.741.824$	10^9
1 terabyte (TB)	2^{40}	10^{12}
1 petabyte (PB)	2^{50}	10^{15}
1 exabyte (EB)	2^{60}	10^{18}



Βασική λογική σχεδίαση και άλγεβρα Boole

Οι βασικοί λογικοί (ή Boolean) τελεστές που χρησιμοποιούνται είναι οι: **NOT, AND, OR**.

Πύλες = Βασικά ψηφιακά λογικά στοιχεία που αντιστοιχούν στις βασικές λογικές συναρτήσεις.

AND: Λογικός πολ/σμός (σύζευξη)

Boolean Έκφραση: **$A \bullet B$**

OR: Λογική πρόσθεση (διάζευξη)

Boolean Έκφραση: **$A + B$**

NOT: Λογική αντιστροφή

Boolean Έκφραση: **\bar{A} ή $\sim A$ ή A'**

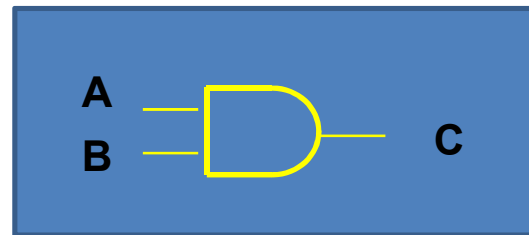
Λογικός πολλαπλασιασμός (AND)

Έχει αποτέλεσμα = 1, μόνο εάν και οι δύο είσοδοι είναι 1.

TRUE=1, FALSE=0

$$C = A \cdot B \text{ ----- } C = A \text{ AND } B$$

Λογική Πύλη



Πίνακας αληθείας

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

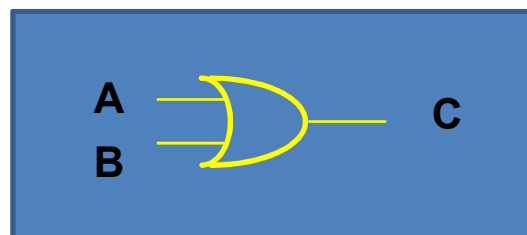
Λογική πρόσθεση (OR)

Έχει αποτέλεσμα = 1, εάν κάποια από τις εισόδους ισούται με 1.

TRUE=1, FALSE=0

$$C = A + B \text{ ----- } C = A \text{ OR } B$$

Λογική Πύλη



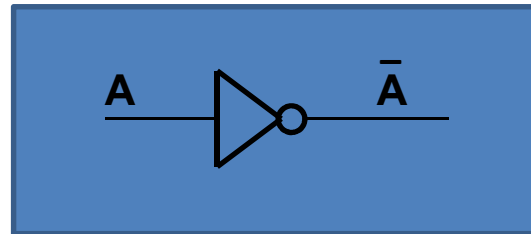
Πίνακας αληθείας

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Λογική αντιστροφή (NOT)

Έξοδος αντίστροφη της εισόδου

Λογική Πύλη

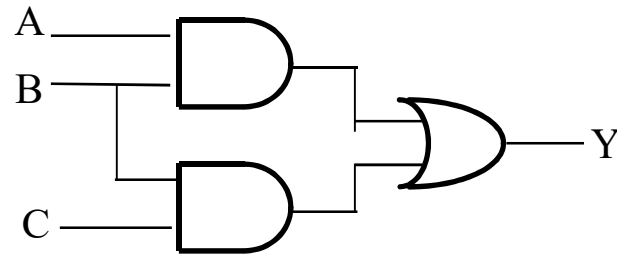


Πίνακας αληθείας

A	$\sim A$
0	1
1	0

Συνδυάζοντας Λογικές Πύλες

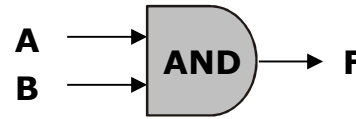
Η έξοδος ενός συνδυαστικού κυκλώματος εξαρτάται μόνο από τις εκάστοτε εισόδους.



AND - OR Λογικό Κύκλωμα

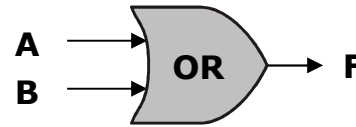
Σύνοψη: Λογικά κυκλώματα - Πύλες

- AND: $F = A \cdot B$



Η έξοδος F είναι 1 όταν **όλες** οι εισοδοι (A, B) είναι 1

- OR: $F = A + B$



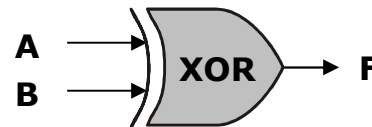
Η έξοδος F είναι 1 όταν **τουλάχιστον** μία εισοδος είναι 1

- NOT: $F = A'$



Η έξοδος είναι το αντίθετο της εισόδου

- XOR: $F = A' \cdot B + A \cdot B'$



Η έξοδος F είναι 1 όταν **μία και μόνο μία** εισοδος είναι 1

- NAND: $F = (A \cdot B)'$
- NOR: $F = (A + B)'$
- XNOR: $F = A \cdot B + A' \cdot B'$

Πύλες: Άσκηση

Έστω η λογική συνάρτηση:

$$F = A \cdot B' + A' \cdot B \cdot C$$

1) Γράψτε τη συνάρτηση, χρησιμοποιώντας τα ονόματα των πυλών (AND, OR, NOT)

2) Σχεδιάστε το αντίστοιχο λογικό κύκλωμα

Πόσες πύλες θα έχει;

Πόσων εισόδων η κάθε μία;

3) Σχεδιάστε τον πίνακα αληθείας της F

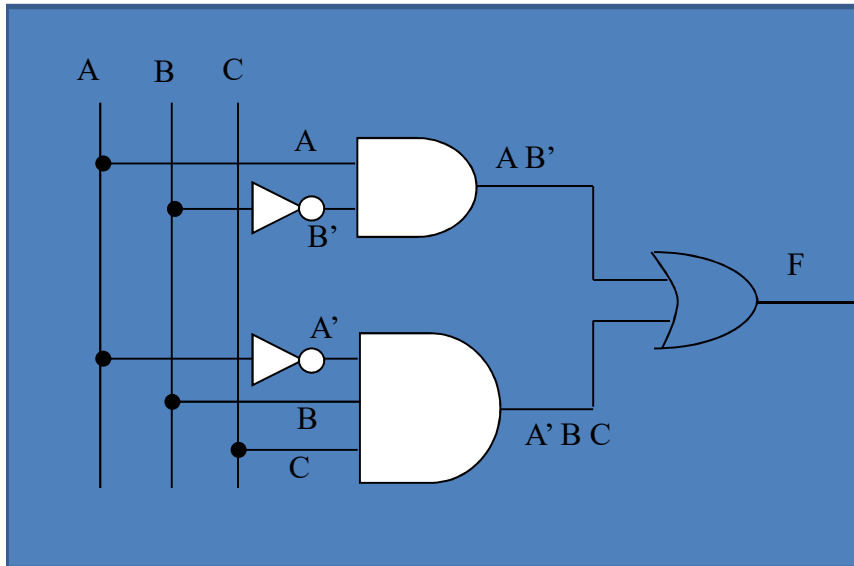
Πόσες γραμμές και πόσες στήλες θα έχει;

Πύλες: Άσκηση

$$F = A \cdot B' + A' \cdot B \cdot C$$

$$F = (A \text{ AND } (\text{NOT } B)) \text{ OR } ((\text{NOT } A) \text{ AND } B \text{ AND } C)$$

Λογικό κύκλωμα



Πίνακας αληθείας

A	B	C	A'	B'	K = AB'	L = A'BC	F = K+L
0	0	0	1	1	0	0	0
0	0	1	1	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	1	1
1	0	0	0	1	1	0	1
1	0	1	0	1	1	0	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Εφαρμογή λογικών πράξεων σε ομάδες bits: Quiz

Πόσο κάνει (ΣΤΑ 4 bit)

- 3 AND 5
- 7 AND -4
- NOT 3
- -2 OR 1
- -2 XOR -1

Πόσο κάνει (ΣΤΑ 8 bit)

- 3 AND 5
- 7 AND -4
- NOT 3
- -2 OR 1
- -2 XOR -1

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

A	$\sim A$
0	1
1	0

Ερωτήσεις?



Thank you
for your
attention!

Ερωτήσεις Ανακεφαλαίωσης

1. Ποια είναι η μορφή του αριθμού $(11010)_2$ στο δεκαδικό;
2. Ποια είναι η μορφή του $(152,35)_{10}$ στο δεκαεξαδικό;
3. Μετατρέψτε τον $(F3E,F2)_{16}$ στο δυαδικό
4. Πώς αναπαρίσταται το -5 σε παράσταση συμπληρώματος ως προς 2 στα 8 bit;
5. Ποιον αριθμό αναπαριστά η ποσότητα 1001 αν μιλάμε για
Μόνο θετικούς αριθμούς / Παράσταση μέτρου / ΠΣ2
6. Ποιος είναι ο πιο μεγάλος αριθμός που μπορώ να παραστήσω στα 6 bit με ΠΣ2;
7. Συντάξτε τον πίνακα αληθείας της λογικής συνάρτησης
$$F = A.B + C'$$